Applicability of Dynamical Modelling and Theoretical Control Methods in Tree Growth Prediction and Planning

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy (Forestry).

Oliver Chikumbo

Department of Forestry

Australian National University

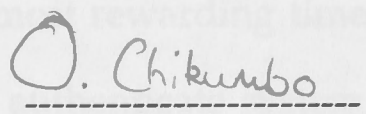
Canberra, ACT 0200, Australia

June 1996

But many of the priests and Levites and chief of the fathers, who were ancient men, that had seen the first house, when the foundation of this house was laid before their eyes, wept with a loud voice; and many shouted aloud for joy:

Ezra 3:12, -Holy Bible (KJV)

Except where otherwise indicated this thesis is my own work.



Oliver Chikumbo

Acknowledgments

Many thanks to my mum Regina Chikumbo and brother-in-law, Steve Moyo for their overwhelming support at the start of my research program. I am also grateful to my supervisors, Dr. Brian Turner and Dr. I.M.Y. Mareels, who had ample time to listen and to give me exceptionally constructive advice and a myriad of references to read. This helped me to clearly define my research problem.

Acquiring data proved to be a major problem and through Dr. Ryde James (Department of Forestry, Australian National University), I managed to secure data from the Forest Research Institute and Tasman Forestry (New Zealand) and I am really grateful to him. Many thanks to Stuart Christie (Forestek-CSIR, South Africa), who made it possible for me to work with Heyns Kotze (Forestek-CSIR, South Africa), which proved to be one of my most rewarding time in my research career. Through Forestek I secured data to fully authenticate system identification and optimal stand density control.

I am also grateful to the Department of Natural Resources and Environment (DNRE) and especially to Fiona Hamilton and Bruce Kilgour who managed to secure a job for me, so that I could complete my research. I am now employed on a full-time basis as a Forest Biometrician for DNRE. I am also thankful to my friends, James Allnutt, James and Karen Nachipo, Deb O'Connell, Carol and Martin Mutendeudzi and Florence Soriano who stuck to me like family during my research. Many thanks to Roland Jahnke (Department of Forestry, Australian National University) for assisting me to acquire MatLab from Ceanet (Sydney).

Finally, I thank the Almighty God and Lord Jesus Christ for granting me the wisdom to rely on Him in everything and granting me the ability and willingness to accept help from other people.

Abstract

Determining optimal thinning strategies on a stand basis for forest plantations is a problem that forest analysts have solved with multi-stage optimisation procedures based on dynamic programming ideas. Recent literature suggests that the Principle of Optimality is violated by the forestry formulations of dynamic programming - this is the neighbourhood storage concept that limits exhaustive search of a solution at each decision stage, followed by a forward recursive optimisation. This particular formulation has been largely driven by an attempt to overcome the 'curse of dimensionality' forced on by computer memory limitations. Another problem which confronts the analyst is the absence of suitable forest growth models (ones directly related to the decision variable).

In this thesis, models based on control theory are used successfully to solve the optimal thinning strategy problem. These models are nonlinear dynamical (input/output) models (Systems Engineering terminology used to describe dynamic systems, where control and controlled variables are identified). To fully account for the nonlinearities in the observed growth trends, a 2-stage modelling approach is adopted; in the first stage a linear dynamical model, acounting for the general growth trend, is identified and in the second stage the parameters from the model in the first stage are modelled as functions of initial densities, thus resulting in a nonlinear dynamical model with good predictive properties; the growth dynamical models developed for *Pinus patula* stands were significantly better in prediction than previously published growth models.

A state space control model is developed (from these dynamical models) with an input/state/output structure that makes it possible to direct a forest stand to achieve the desired growth performance. The input represents the control variable (expressed in stems/ha of the number of trees to be cut); the state (expressed as a

vector of the number of standing trees (stems/ha)), stand basal area (m^2/ha) and mean stand height (m)) represents sufficient growth information to predict the future output; and the output (expressed as stand volume in m^3/ha) represents the growth response.

In addition to this control model a cost functional is formulated to maximise volume or value production over a specified rotation length. A control sequence is then determined by using an iterative solution technique such as dynamic programming or the maximum principle.

The silvicultural strategies determined from the control model were comparable with those determined from long-term analysis of trial experimental data; an indication of the reliability of the control model. In the event of changes in market forces, political emphasis, climate or site productivity, the control model can be reliably used to promptly determine alternative strategies by manipulating the constraints and/or the cost functional.

Growth data from Australia (Victorian Department of Natural Resources and Environment), New Zealand (Forest Research Institute and Tasman Forestry) and South Africa (Forestek-CSIR) were used. The data were from three types of forests namely the *Eucalyptus* mixed-species forests (Victoria, Australia), *Pinus radiata* plantation forests (New Zealand) and *Pinus patula* plantation forests (South Africa).

MatLab software is used for developing dynamical models and a FORTRAN program called DMISER3, is used to find solutions to the optimal control formulations.

TABLE OF CONTENTS

APPENDIX I

APPENDIX II

APPENDIX III

CHAPTER 1

Introduction

Planning is the problem of determining an optimal procedure for attaining a set of objectives (Luenberger, 1969). In timber production, planning has been primarily concerned with 'forest regulation' based on the desirability of attaining some 'target forest' structure (Clutter, et al., 1983). Forest regulation was a harvesting control that was determined from timber inventories conducted to find out the total volume in a forest. This information was used in models to calculate a sustainable yield for the forest, based on average forest-wide growth rates (Turner, 1995). Each year the sustainable yield had to be removed from the forest after a thorough check of maps to determine where the cut could be taken from. Usually this was done only for the first few years; rarely was a check done to see whether at the end of the cutting cycle or rotation it would be possible to begin again (Turner, 1995).

However, the objectives have changed in recent years and are more concerned with immediate, rather than distant future, characteristics of the forest. Planning the management of a plantation forest, which is the focus of this thesis, now consists of defining operational area units and estimating the growth and yield for each unit under a set of alternative activities (harvesting/silvicultural practices). Planning is the most important level of control in any forest enterprise because it is primarily through planning that a high level manager exerts influence over his or her organisation. Accurate estimations of growth and yield provide a basis for determining the 'best' options for maximising management objectives; computer decision support systems built from growth and yield models are used to evaluate the different forest management options for bare land or existing forest stands. These options are ranked in terms of volume or value production criteria and optimum planting densities, *thinning*[1] regimes or rotation ages can be determined.

---

[1] Thinning is an important silvicultural treatment that influences growth through competition control.

The growth and yield models describe different components of plantation growth and may include models for dominant height growth, survival, basal area growth, volume growth, diameter distribution and height by diameter class (Harrison, et al., 1994). A forest planning decision support system may incorporate a mathematical programming formulation because a choice has to be made between alternative options. Optimisation techniques have been and are still being used to determine a schedule of activities to satisfy management objectives. An important subclass of mathematical programming formulations is dynamic programming.

Problems which require sequential decision-making in forest planning (or any other discipline) can be dealt with effectively if considered in a multi-stage optimisation framework. In a multi-stage optimisation formulation the outcomes at each and every stage are calculated on the basis of the Principle of Optimality (Bellman, 1957) that basically states that whatever the first decision is for an optimal policy, the remaining decisions must constitute a policy with regard to the state resulting from the first decision.

Various tools are available to find solutions to multi-stage optimisation problems. Dynamic programming and maximum principle are the two solution techniques that have received the most attention in the last half-century. The availability of powerful computers and appropriate software, makes it possible for many carefully formulated multi-stage optimisation problems to be solved. A major break through in the mathematical formulations came with the introduction of state space representation by Rudolf Kalman in 1960.

State space representation (see chapter three) defines a direct functional relationship between the control variables and the equations that simulate the motion (behaviour) of a system and these equations describe the state or the 'memory' of the system. A state is a set of descriptive variables that provide all the information about the past necessary to make immediate and future decisions based on the state

equations and the present and future control variables. The equations of motion are mathematical approximations of how the state variables change in response to management actions (control variables). The success of such a mathematical formulation (state space representation) lies in the use of very 'simple' model structures, called 'dynamical models' that enable one to model (only) the variables of concern. Dynamical models are parametrically efficient in that only the lagged output variable (response variable) and the input variable (control variable) constitute the model structure (see chapter three). The models as such tend to have very good statistical properties compared to multiple regression models. Multiple regression models can be difficult to control by virtue of the number of explanatory variables which themselves may have to be predicted, thereby increasing the estimation error.

Dynamical models can be used in conjunction with a cost functional to determine a sequence of control inputs that would yield an optimal performance. In the context of a thinning problem, the cost functional would be characterised by an economic return of a forest plantation.

The particular problem with dynamic programming formulation that Chen et al., (1980) cited, was the need to define all the possible states at each stage and this can be computer-memory demanding. To counter the problem foresters have developed a dynamic programming formulation that requires user-specified possible states that are common to all the finite stages in the planning period (Buongiorno and Gilless, 1987). For brevity this formulation will be referred to as 'forestry dynamic programming'. The forestry dynamic programming formulation has not taken advantage of the state space representation and relies on specifying 2 to 3 states (Garcia, 1990) at each stage. The choice of these states are based on expert knowledge of the forests in question. The stages are normally specified at equal intervals of 5 to 10 years. This kind of formulation is very limiting in that decisions are confined to long interval points and there is no exhaustive search of the states at each stage. Therefore,

there is no guarantee of finding a global optimum over the planning period. With the forestry dynamic programming formulation, one can still obtain a sub-optimal solution and there is no easy way of checking whether the solution is sub-optimal or optimal. The forestry dynamic programming formulation can be described as heuristic methods combined with recursive optimisation.

In contrast, dynamic programming and the maximum principle will search all possible states as constrained by the equations of motions (see appendix II). The maximum principle satisfies a necessary and sufficient condition (for an optimal path to give a minimum cost) that a decision be chosen such that the *Hamiltonian*[2] takes the maximum possible value at each stage of the path (Boltiyanskii et al., 1962). Dynamic programming employs the imbedding technique which enables an exhaustive search of all the possible states at each stage in a backward recurrence mode (see appendix II). Some examples of forestry dynamic programming are given in this chapter to set the stage for the definition of the problem addressed in this thesis.

1.1 Recent 'forestry dynamic programming' formulations

Haight et al., (1985) developed a forestry dynamic programming algorithm for determining thinning regimes for lodgepole pine (*Pinus contorta* Doug. ex Loud).

---

[2] The Hamiltonian is a function H such that a given PARTIAL DIFFERENTIAL EQUATION of first order can be rewritten as

$$\partial u / \partial t = -H(t, x_1, \ldots, x_n, p_1, \ldots, p_n),$$

where the variables are all functions of the parameter t (Borowski and Borwein, 1989). This is a *Hamilton-Jacobi* type differential equation. The Hamiltonian exists for any equation

$$F(x_0, x_1, \ldots, x_n, u, p_1, \ldots, p_n) = 0,$$

where $p_k = \partial u / \partial x_k$, that does not depend on u. The *Hamiltonian canonical form* is then

$$\frac{dx_k}{dt} = \frac{\partial H}{\partial p_k}, \frac{dp_k}{dt} = -\frac{\partial H}{\partial x_k}.$$

The stages were set at twenty year intervals. The formulation had two decision variables, thinning type and residual number of trees. A node represented a state which was a four-element vector consisting of stand age, residual number of trees, residual basal area and thinning type. To start the calculations, the following stand information was provided: age, site index and a list of tree diameters and heights. A simulator proceeded by updating the tree attributes annually and removing trees that died from the list.

At a decision stage, all the thinning types were applied with a predetermined intensity of harvest: thinning from below (trees removed from the classes with the smallest diameters); mechanical thinning (a proportion of trees removed from each tree class); thinning from above (trees removed from the tree classes with the largest diameters); and no thinning. The resulting stand structure was then used to calculate the present net worth (PNW) by using a stumpage value of the harvested material of average diameter at breast height. The algorithm employed a forward recursion, calculating at each stage a cumulative PNW and storing in computer memory the highest value. The calculations terminated at a prescribed age and the attributes of the stand with the 'optimal' PNW retrieved. Admittedly Haight et al., (1985) pointed out the weakness of their methodology that the thinning sequence selection was not based on an exhaustive search for an optimal state at each stage.

Filius and Dul (1992) investigated the impact of stumpage price on the rotation and thinning strategies of Douglas-fir (*Pseudotsuga menziesii* (Mirb) Franco) by using a forestry dynamic programming formulation. It was based on a forward recurrence procedure with a three-state descriptor (a node) that consisted of age of the stand in years, the basal area per hectare and the number of trees per hectare. Filius and Dul (1992) limited their number of nodes at each stage by using the 'neighbourhood storage location' technique. For example, a state of 380 trees with a basal area of 23.2m^2 at age 30 years belonged to the 'neighbourhood' of the node

defined as the interval 300-400 trees and 22-24m^2 basal area at an age of 30 years. Basal area to be removed was used as a decision variable and was related to the interval of the state descriptor basal area. The authors managed to come up with a solution to their problem (impact of stumpage price on rotation and thinning strategies) and speculated on further developments of their algorithm such that it could take into account thinning types, as in the previous example of Haight et al., (1985). The inability to search through all the possible states was a limiting feature of the formulation.

Anderson and Bare (1994) also developed a forward recursion, discrete two-state (number of trees per hectare and basal area per hectare), forestry dynamic programming problem that maximised the PNW of harvested trees at each stage. The neighbourhood storage concept was employed where a node represented a two-state descriptor. It was not clear how the decisions were made on the harvest removed. The residual stand structures were grown between nodes by using growth functions and the states classified into neighbourhood storage classes. The stand structures possessing the largest accumulated PNW at each stage were chosen as the 'optima'. The process was repeated until the last stage was reached and stand structures retrieved to form the 'optimal' trajectory.

Anderson and Bare (1994) compared their PNW results to those obtained by Haight et al., (1985) and concluded that solutions produced by these optimisation techniques were only local and not global. However they attributed the problem to ill-behaved growth functions. They also suggested that the original work by Adam and Ek (1974), on optimisation of uneven-aged hardwoods, may have resulted in local optima.

Arthaud and Klemperer (1988) concluded that the neighbourhood storage combined with a forward recursion optimisation resulted in violation of the Principle

of Optimality; i.e. this formulation led to loss of paths that would have been optimal over the long run, but were discarded because of lower values (generated through thinning). These few examples illustrate the current concept of forestry dynamic programming and by purely observing the results, some authors (e.g. Anderson and Bare, 1994; Arthaud and Klemperer, 1988; Arthaud and Warnell, 1994; Pelkki, 1994; Valsta, 1994) have found out that they may be just achieving local optima. In appendix II more theoretical background and the mathematical formulation of forestry dynamic programming are given, that will enhance an understanding of its main differences from the dynamic programming formulation developed from control theory.

1.2 Aims

To redress the local optima and computer memory problems of forestry dynamic programming formulation, appropriate stand level functions have to be developed. These functions, integrated in a dynamic programming structure, enable an exhaustive search for the optimum at each stage in the planning period. These are the types of problems commonly addressesd by systems engineers in controlling industrial systems. Thus, the aims of this research are to use a systems engineering approach to:

(a) identify dynamical model structures that describe tree stand growth *trends* (dynamics of a system) and are parametrically efficient;

(b) demonstrate how to adapt model structures in (a) to different sites;

(c) demonstrate the development of a suitable diameter distribution model that will partition growth estimated in (a) into diameter classes; such a model could be used in (d) for projecting the growth in diameter classes over the rotation

period and help redirect the thinning procedure to the appropriate diameter classes; and

(d) develop a stand level control system (based on systems theory) that will assist in determining an optimal rotation and thinning strategy by using dynamic programming or maximum principle formulation.

1.3 Theme and chapter outline

The concepts of system identification are covered in chapter two. There is a rudimentary introduction to control theory and its link to system identification. In chapter three specific model structures determined from system identification are given, the focus being on linear time-invariant models. A case study on modelling dynamical models for forest stand growth and the formulation of a thinning multi-stage optimisation problem are found in chapters four and five respectively. In chapter six the use of *recursive identification* to fine tune a 'guessed' growth model from a family of growth curves is demonstrated, with its emphasis on extending dynamical models to different site qualities in a region. Such techniques (recursive identification) hold the key to extending the control formulation in chapter five beyond the stand level modelling. Chapter seven is about possible extensions of system identification applications to other forest related problems and speculations are made on the future developments of an economic component that can be incorporated in the control model formulation of chapter five. The dissertation concludes with chapter eight that is a summary of the findings of this research work.

Details of the tools used and mathematical formulations are found in appendices I-III.

CHAPTER 2

System identification

Forest management is the art and science of making decisions about the use and organisation of forests. Such decisions may involve the long-term future of the forest or the day-to-day activities. Given the multi-dimensional nature of the management of forests, the ability to make sound management decisions hinges on the use of decision support systems. Correct use and interpretation of results from a decision support system will ensure optimal or near-optimal decisions. The problem that is exclusively dealt with in this research is the development of a decision support system that can be used to determine and evaluate various silvicultural thinning strategies for forest plantations, and the literature review covers the concepts used to develop this system.

These concepts of 'systems' theory apply to everyday situations and are simple and application-independent. Systems theory describes an abstract situation called a system that is characterised by a set of elements connected by *information links* within some delineated system boundaries (Leigh, 1992). The system boundary is not a physical boundary but rather a convenient abstract device. In this dissertation, a forest plantation is considered as a system characterised by the following elements: stand density; stand basal area; stand height; and stand volume. These elements (or variables) are connected (via mathematical functions - the information links) in such a way as to form and/or act as an entire unit, that is 'the system'.

All techniques for analysis (or investigation of the properties of a system) and design (i.e. the choice and arrangement of system elements to perform specific tasks) of systems are based on the availability of appropriate functions that model the process dynamics. The art of analysing a system was facilitated by the founding of *system identification* methods. There are many textbooks on the topic of system

identification and Professor L. Ljung gives a coherent coverage of this topic in his textbook (Ljung, 1987) that is highly recommended for further reading on system identification methods. A specific area of these methods is covered in this chapter that is applicable to the models developed in chapter four for the control model in chapter five.

System identification methods enable the determination of dynamical models that are the mainstay of analysis and design of systems. The development of dynamical models of physical systems require 80 - 90% of the total effort spent in system analysis and design (Phillips and Harbor, 1991). In certain cases, laws of physics can be used to obtain a system model for analysis. Some biological systems, for instance, obey the second law of thermodynamics that describes all events involving energy exchange (Raven, 1982).

2.1 System Identification

System identification is the problem of building mathematical models ('dynamical models') of a dynamical system based on input and output observations from that system (Ljung, 1991). By definition an input variable is one that can be measured and controlled by the observer. The output variable can also be measured but the observer has no direct control; it is a response to the combined effect of the input and the external stimuli (disturbance or noise) that act on the system (see Figure 2.1).

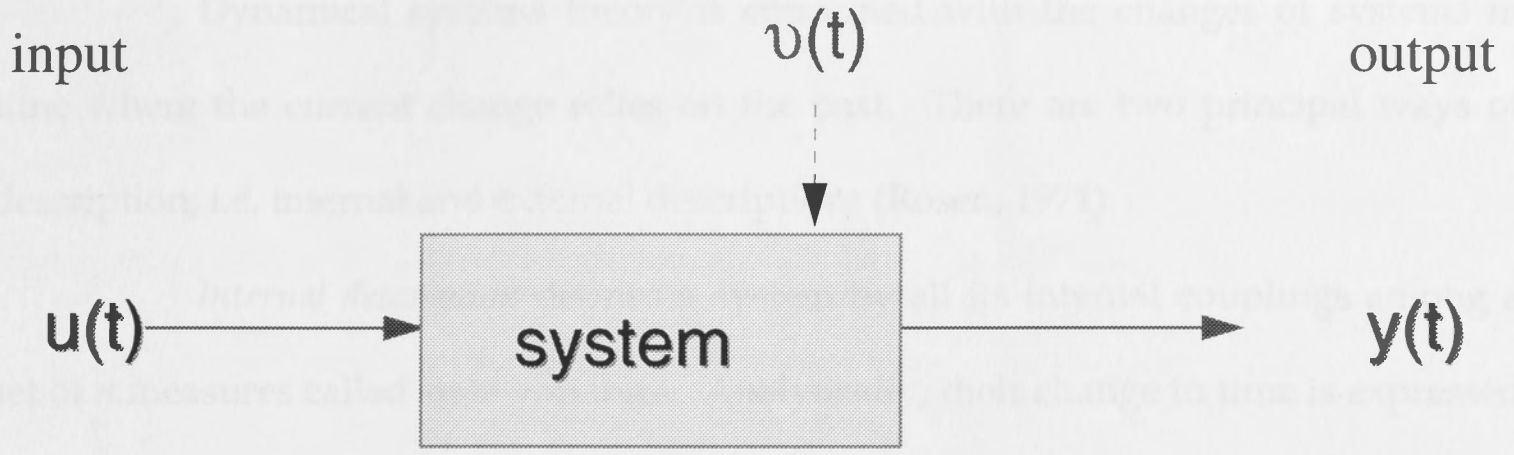


Figure 2.1: A system showing the relationship of input, $u(t)$, and disturbance, $v(t)$, to the output, $y(t)$.

For the thinning problem, the input is the number of trees cut, and the output is the stand basal area, stand height and standing trees. The disturbance is characterised by climate, fire, fertilisation etc.

2.1.1 Dynamical Systems

A dynamical system is a system that evolves with time and can be described as a triple Σ:[2] = $\{T, W, \beta\}$ with T the time set, W the variable responses, and β $\subset W^T$ the behaviour of the system (Willems, 1986a). T represents the set of time (or sampling) instances that are relevant to this system: usually T is an interval in Z [3]. The signal alphabet W represents the space in which the variables of interest, through which the system interacts with the environment, take on their values. The elements of W constitute the attributes of the system. The behaviour, β, consists of those trajectories $w : T \rightarrow W$ which are compatible with the laws governing the dynamical system.

---

[2] Σ expresses the laws among the variables i.e., the compatibility relations which the different variables, in order for them to exist simultaneously, need to satisfy.

[3] Unless otherwise specified:

Z = the set of integers: ..., -2, -1, 0, 1, 2, ...

Dynamical systems theory is concerned with the changes of systems in time where the current change relies on the past. There are two principal ways of description, i.e. internal and external descriptions (Rosen, 1971).

Internal description defines a system by all its internal couplings among a set of n measures called state variables. Analytically, their change in time is expressed by a set of n simultaneous first order difference equations (or differential equations for continuous-time systems) called dynamical equations or equations of motion of the system (Bertalanffy, 1973).

Geometrically, the change of the system is expressed by the trajectories the state variables traverse in the state space, i.e. n = dimensional space of possible location of the state variables. Three types of behaviour may be distinguished and defined as follows (Bertalanffy, 1973):

(a) A trajectory is called *asymptotically stable* if all trajectories sufficiently close to it at time, $t = t_0$ (initial time) remain close to it and approach it asymptotically when $t \to \infty$;

(b) A trajectory is called *neutrally stable* if all trajectories sufficiently close to it at $t = t_0$, remain close to it for all later time but do not necessarily approach it asymptotically; and

(c) A trajectory is called *unstable* if at least one of the trajectories close to it at $t = t_0$, do not remain close to it as $t \to \infty$.

A central notion about dynamical theory is that of stability, i.e. response of a system to perturbation (Bertalanffy, 1973). The concept of stability originates in mechanics (that a motion is stable if insensitive to small perturbations) and is generalised to the 'motions' of the state variables of a system.

In *external description*, the system can be considered as a 'grey box' or as a 'black box' depending on whether the physics of the system are known or not. Model

sets that have adjustable parameters with physical interpretation are called grey boxes. In some cases standard models may be employed without reference to the physical background. Such model sets whose parameters are basically viewed as vehicles for adjusting the fit to the data and do not reflect physical considerations in the system, are called black boxes (Ljung, 1987). A black box system description is typically expressed in terms of input and output variables, via difference (or differential) equations.

In principle it is usually possible to develop an internal description from an external one and vice versa. In the case of linear systems formal techniques to achieve this end are established (Willems, 1986a). When dealing with the general nonlinear systems, such transformations are nontrivial. One moves from an internal description to an external one by eliminating the state variables from the system equation. By appropriately identifying a state variable defined via external variables the external description can be transformed into an internal description.

In the context of system identification in particular where the physical laws describing the system are too complex, external descriptions are easily obtained and may be derived from data directly. It is essential to ensure that system identification techniques lead to approximate system representations where the quality of a model, depending on the objective of the modelling exercise, is expressed by its predictive, simulation or control capabilities.

2.1.2 System Identification procedure

A dynamical model in discrete-time may be mathematically expressed as a difference equation. In practice, circumstances usually do not allow an exact mathematical representation of a system. However, if valid assumptions are made on the system properties, a great deal of valuable information can be gained (Kuo, 1962). To clarify the statement just made, it should be realised that all dynamical systems are nonlinear to some extent and the mathematical treatment of nonlinear systems is

extremely difficult. Therefore, it is often necessary to assume that the system under study behaves linearly over a range of operational conditions. The accuracy (or closeness to the truth) of the models can be improved by increasing the complexity of the equations, but exactness (absolute truth) cannot be achieved. Therefore, it is important to strive to develop a model that is adequate for the problem at hand without making it overly complex. In some cases the assumption of linearity may depart greatly from reality.

The procedure to determine a model of a dynamical system from observed input/output data involves three basic ingredients:

(a) the (input/output) data;

(b) a set of candidate models (the model structures); and

(c) a criterion to select a particular model in the set, based on the information in the data (Ljung, 1991).

The input/output data are sometimes recorded during a specifically designed identification experiment, where the user makes a choice on what to measure (signals) and when to measure (sampling instants) over some sampling interval. The object with experimental design is thus to make these choices so that the data becomes maximally informative. The key to success lies in:

(a) obtaining good quality data;

(b) a good knowledge (of the physics and/or ecology) of the system; and

(c) having a good feel for the character of the model structure that should be used (Ljung, 1987).

Without a reasonable data record not much can be done and there are several reasons why such a record cannot be obtained in certain applications. A quite obvious reason is that the time scale of the process is so slow that any informative data records by necessity are short. Ecological and economic systems clearly suffer from

this problem. Another reason is that the input may not be open to manipulations. Even if the manipulation of the input and measurement for long periods is possible, it may be difficult to obtain a good data set. The prime reason for this is the presence of unmeasurable disturbances that do not fit well into the standard picture of stochastic processes.

The set of candidate models is obtained by specifying from which collection of models[4] to look for a suitable one. This is the most important and most difficult part of the system identification procedure because it calls for:

(a) a priori knowledge and mathematical intuition; and

(b) an insight into modelling, e.g. constructing a model with some unknown physical parameters from basic laws of nature and other well established relationships or applying standard linear models without reference to the physical background of the system.

The 'best' model in the set is determined by applying diagnostic checks with the object of uncovering possible lack of fit and diagnosing the cause. This can normally be achieved by checking the mean squared error on how predictive the model is against fresh data that were not used to construct the model. A bad choice of a model structure cannot offer a good model, regardless of the amount and quality of the data. Some processes will admit standard, ready-made (black box) model descriptions and some, tailor-made model sets. In the latter, some physical insight is required before a model can be estimated. This problem is clearly application dependent.

The identification process involves searching for a model structure by computing the best model in the set and evaluating its properties to see if they are satisfactory. The process can be listed as follows:

---

[4] The collection of models is given in chapter three.

(a) Design an experiment and collect input/output data;

(b) Examine the data; remove trends[5] and outliers;

(c) Select and define a model from a set of candidate system descriptions within which a model is to be found;

(d) Compute the best model according to (a) and a given criterion of fit;

(e) Examine the obtained model's properties by checking against fresh data; and

(f) If the model is good enough, then stop; OR go back to (c); OR go back to (d); OR work further on (a) and (b).

It cannot, however, be sufficiently stressed that the key to successful modelling lies in thinking, intuition and insight; these cannot be made obsolete by automated model estimation.

2.2 The Linear Model Description

This section describes the way of mathematically defining the behavioural patterns of linear dynamical systems. It is only when the dynamic characteristics of a system are understood that intelligent direction, manipulation and control of the system is possible. Concentration is on causal time-invariant linear systems, that were found to be of great importance in deriving models that could accurately describe suitable forest growth trends (Chikumbo et al., 1992; Chikumbo et al., 1993; Chikumbo and Mareels, 1993; 1995a; 1995b). The growth models developed in this thesis were nonlinear, but were characterised by linear time-invariant model structures as the

---

[5] Care should be taken in removal of trends or detrending as this is applicable only to special types of data. An example included in Endnote 2A illustrates the error that can result from inappropriate use of detrending.

main building blocks. The following procedure was successfully adopted in constructing the growth models:

(a) recognise linear-like responses and model these using time-invariant linear model;

(b) recognise the critical system parameters that influence the responses from the model in (a) and estimate them as linear regression or polynomial functions;

(c) fit the functions from (b) to the model in (a) to give a robust model.

Consider a system with a scalar input signal, $u(t)$; $t = 1, 2, ..., N$, and a scalar output signal $y(t)$. The system is said to be *time-invariant* if its response to a certain input signal does not depend on absolute time (Ljung, 1987). It is said to be *linear* if its output response to a linear combination of inputs is the same linear combination of the output responses of the individual inputs.

Introduction of the transfer function and impulse response concepts assists in understanding linear system description. The modes of description are related and each offers advantages and disadvantages in different fields of application and circumstances (Box and Jenkins, 1971).

Let $y(t)$ be represented as a function of the past values of $u(t)$ and noise $v(t)$, thus:

$$y(t) = g_0u(t) + g_1u(t-1) + g_2u(t-2) + ... + v(t) \tag{2.1}$$

or in shorthand,

$$y(t) = G(q)u(t) + v(t) \tag{2.2}$$

where $G(q) = g_0 + g_1q^{-1} + g_2q^{-2} + ...$, and q^{-1} is the backward shift operator defined as $qu(t) = u(t-1)$ and $v(t)$ is a random variable with mean zero, a fixed covariance structure and independent of $u(t)$, $u(t-1)$, $u(t-2)$, ... The weights g_0, g_1, g_2, ... are called

the *impulse response weights* and a graph of these weights is called an *impulse response function*.

Suppose that u is held indefinitely at some value, and the total change in y is observed; the change in $y(t)$ represents the sum of the impulse response weights, g_k, $k = 0, 1, 2, \ldots$. Therefore, this imposes a restriction that

$$G(1) = \sum_{k=0}^{\infty} g_k < \infty \tag{2.3}$$

The value $G(1)$ is the *steady state gain* of the system and it represents the total change in $y(t)$ for a unit change in $u(t)$ held indefinitely at some new value (Vandaele, 1975). The unit change in $u(t)$, is a *step* change and the response that results from a step change is an impulse response for a continuous-time system or pulse response for a discrete-time system. A system that satisfies (2.2) is said to be stable, if the infinite series $g_0 + g_1q^{-1} + g_2q^{-2} + \ldots$ converges for $|q| \geq 1$. Stability implies that a finite change in the input results in a finite change in the output. Thus for a stable system the sum of the impulse response weights converges and is equal to the steady state gain of the system (Box and Jenkins, 1971).

Under general conditions $G(q)$ can be either approximated or exactly represented by a ratio of two finite polynomials in q:

$$G(q) = \frac{B(q)}{A(q)} \tag{2.4}$$

where $B(q) = b_0 + b_1q^{-1} + \ldots + b_mq^{-m}$ (2.5)

$$A(q) = a_0 + a_1q^{-1} + \ldots + a_nq^{-n} \tag{2.6}$$

Without loss of generality the coefficient a_0 can be normalised as unity. Equation (2.2) thus becomes:

$$y(t) = \frac{B(q)}{A(q)}\ u(t) + v(t) \tag{2.7}$$

It is possible that a delay or *dead time*, in the system's response can be present. To take into account this generality τ, the dead time, is introduced to indicate the number of periods it takes before $u(t)$ starts influencing $y(t)$. The transfer function model (2.7) may then be written as:

$$y(t) = \frac{B(q)}{A(q)} u(t - \tau) + v(t) \tag{2.8}$$

In equation (2.2), $v(t)$ can often be described as filtered white noise[6] :

$$v(t) = H(q)e(t) \tag{2.9}$$

where $e(t)$ is white noise and

$$H(q) = \sum_{k=0}^{\infty} h(k)q^{-k} \tag{2.10}$$

Noise can be a result of drift of the sensors that measure the signals or the influence of signals that have the character of inputs but are not controlled by the observer. The numbers $\{h(k)\}$ are transfer functions from e to y. Equations (2.2) and (2.9) combined, will thus become:

$$y(t) = G(q)u(t) + H(q)e(t) \tag{2.11}$$

2.3 Modelling for Simulation, Prediction and Control

Objectives of a modelling exercise should be made clear at the outset so that the effort spent in developing a model of a system is related to the application it is intended for. After objectives have been set, selection of the most appropriate type of model to achieve these objectives is carried out and decisions are made on the type of modelling approach to use. This section considers three approaches in modelling namely *simulation*, *prediction* and *control*. Throughout this section the system description is assumed to take the form of equation (2.11).

2.3.1 Simulation

Responses to various input scenarios $\{u*(t),\ t = 1, 2, ..., N\}$ are simulated. The undisturbed output will be as follows:

$$y*(t) = G(q)u*(t),\ \ t = 1, 2, ..., N \tag{2.12}$$

To evaluate the disturbance influence, a random generator is used to produce a sequence of numbers $e*(t),\ t = 1, 2, ..., N$, that can be used as a realisation of a white noise stochastic process with variance l. Hence

$$v*(t) = H(q)e*(t) \tag{2.13}$$

By presenting $y*(t)$ and $v*(t)$, an idea of the system response to $u*(t)$ can be formed (Ljung, 1987).

This way of experimenting on the model (2.11) rather than on the observed physical process to evaluate its behaviour under different conditions (various input scenarios) has become widely used in forestry, e.g. simulating silvicultural regimes that have not been applied before.

2.3.2 Prediction

Consider the descriptions (2.9) and (2.11) and assume that $y(s)$ and $u(s)$ are known for $s \leq t - 2$. Since $v(s) = y(s) - G(q)u(s)$, it also means that $v(s)$ are known for $s \leq t - 2$.

From (2.2) the conditional expectation of $y(t)$ can be given as:

$$\hat{y}(t \mid t - 1) = G(q)u(t) + \hat{v}(t|t - 1)$$

$$= G(q)u(t) + [1 - H^{-1}(q)]v(t)$$

$$= G(q)u(t) + [1 - H^{-1}(q)][y(t) - G(q)u(t)]$$

$$\therefore \qquad \hat{y}(t \mid t - 1) = H^{-1}(q)G(q)u(t) + [1 - H^{-1}(q)]y(t) \tag{2.14}$$

or

---

6 White noise defines disturbance or movement generated by independently distributed random variables {e(t)} with mean 0 and variance λ.

$$H(q)\hat{y}(t\,|\,t-1) = G(q)u(t) + [H(q) - 1]y(t) \tag{2.15}$$

Note that these expressions are shorthand notation for expansions. For example, let $l(k)$ be defined by $\frac{G(z)}{H(z)} = \sum_{k=0}^{\infty} l(k)z^{-k}$, and assume this is well defined for $|z| \geq 1$; $H(z)$ has no zeros and $G(z)$ no poles in $|z| \geq 1$. Thus (2.14) means that

$$\hat{y}(t\,|\,t-1) = \sum_{k=1}^{\infty} l(k)u(t-k) + \sum_{k=1}^{\infty} -\tilde{h}(k)\,y(t-k) \tag{2.16}$$

The equations (2.14-16) make an assumption that the whole data record from time minus infinity to $t-1$ is available. In practice, however, it is usually the case that only data over the interval $[0, t-1]$ are known. The simplest thing to do would be to replace the unknown data (Ljung, 1987). This is achieved by an approximation of the actual conditional expectation of $y(t)$, given data over $[0, t-1]$. The exact prediction involves time-varying filter coefficients and can be computed using the *Wiener-Kalman-Bucy* filter (Athans and Falb, 1966; Barnett and Cameron, 1985; Ljung, 1987).

From (2.11) and (2.14) the prediction error $y(t) - \hat{y}(t\,|\,t-1)$ is given by

$$\boxed{y(t) - \hat{y}(t\,|\,t-1) = -H^{-1}(q)G(q)u(t) + H^{-1}(q)y(t) = e(t)} \tag{2.17}$$

The variable $e(t)$ thus represents that part of the output $y(t)$ that cannot be predicted from past data.

2.3.3 Control-Optimal

When considering physical systems, there are always some definite objectives in the forefront and the aim is to accomplish some given task as 'cheaply' as possible. In other words, the desire is to obtain the 'best' output which makes some cost measure (or performance) large or small. The translation of these objectives into the abstract language of dynamical systems gives rise to what is known as the *control problem* (Athans and Falb, 1966). The essential elements of the control problem are:

(a) a dynamical system to be 'controlled';

(b) a desired output or objective of the system;

(c) a set of *allowable* (or *admissible*) 'controls' (i.e. inputs); and

(e) a *performance* or *cost functional* that measures the effectiveness of a given 'control action'.

The essential elements arise out of the physical control-system design problem. The mathematical model which represents the physical system, has been examined in the preceding sections of this chapter, with the related concepts of input, output and state. Thus in a more general way the problem addresses the *control* of a dynamical system.

In translating a design problem into a control problem, one is faced with the task of describing desirable physical behaviour in mathematical terms. The objective of the system is often translated into a requirement on the output. The control signals in physical systems cannot take on completely arbitrary values but are subject to certain constraints. For example one cannot harvest more than the number of trees in a forest and it is also not practical to harvest just one tree at a given time. Constraints are thus imposed upon the inputs to the system. These constraints lead to a set of admissible inputs (Athans and Falb, 1966; Boltyanskii, 1962).

Frequently the desired objective can be attained by many admissible inputs, and so the designer seeks a measure of performance or cost of control that will allow for choosing the 'best' input. Most of the time the cost functional chosen will depend on the input and the pertinent variables (Elgerd, 1967). When a cost functional has been decided upon, the designer formulates the control problem as follows: Determine the admissible inputs which generate the desired output and which, in so doing, optimise the chosen performance measure. At this point optimal-control theory enters into the picture to aid the designer in finding a solution to the control problem. Such a solution when it exists is called an optimal control.

The word optimum may refer to either a maximum or a minimum depending on the situation at hand (Elgerd, 1967; Maine and Iliff, 1985). The most obvious method of optimisation is an outright search process for all possible solutions in the total set of possible ones until the best one is found. Optimal control design methods utilise, almost exclusively, state variables rather than transfer function system descriptions, and therefore the index of performance as a rule will be expressed as a scalar function, I, of the state and control-force vectors:

$$I = \underset{(t=0,\ \ldots,\ N-1)}{E} \left\{ F_N(x_N) + \sum_{t=0}^{N-1} F_t(x(t), u(t)) \right\} \tag{2.18}$$

where

$x = (x^1(t), x^2(t), \ldots, x^{n-1}(t))$, represents the state; and

$u = (u^1(t), u^2(t), \ldots, u^r(t))$, the control parameters.

The only constraints imposed upon the solution are those that relate to the physical restrictions on the magnitudes of the components of x and u, i.e. $x(t + 1) = F_t[x(t), u(t)]$, $t = 0, 1, \ldots, N - 2$. Thus no a priori assumptions are made as to the system configuration or linearity of control strategy. Translating this formulation to a forestry problem of developing an optimal thinning strategy for a plantation stand, x would be the state vector with state variables such as basal area, height, etc; u, the thinning strategy; F_t, the economic value of the retained crop at time t; and I, the harvesting cost (index of performance).

By observing the magnitude of the constraints, a u solution is sought which optimises the chosen I criterion. The system difference equation. $x(t + 1) = F_t[x(t), u(t)]$, and the prescribed initial and final states, $x(0)$ and $x(t_f)$, must be observed. There are also constraints that are imposed on the optimisation problem (see chapter five).

To recapitulate, a control problem is the translation of a control system design problem into mathematical terms; the solution of a control problem serves as a guide in developing the actual working control system (Athans and Falb, 1966). More on this topic is found in appendix II.

2.4 *Arguments for dynamical models*

Single equation regression models for tree growth are used to interpolate[7] and extrapolate[8] movements in a response variable by relating it to a set of independent variables in an associative framework. Interpolation and extrapolation in dynamical models are based on the dynamics of the response variable (output) that are influenced by the input and the external stimuli.

Consider a time function $g(t)$ which might represent an increment of some tree stand attribute such as average stand diameter. It may or may not be possible to explain (based on the observational parameters, i.e. external stimuli acting on the tree stand system that the observer has no control over, such as drainage, exposure, soil fertility and weather conditions) why $g(t)$ behaved the way it did. Much of the increment over time may have been due to factors that simply may not be explained or accounted for, i.e. data are not available for those explanatory variables that are believed to affect $g(t)$. Even if data for all the variables that influence $g(t)$ are available, ill conditioning (or multicollinearity in the predictor variables) may be a problem.

---

[7] Interpolation means prediction for new cases with independent variables not too different from values of the independent variables in the construction sample (Weisberg, 1985). Interpolation is used synonymously with simulation in this thesis. For a model, interpolation generally occurs when the predictor is in the range observed in the construction sample.

[8] Extrapolation is when the independent variable is estimated out of range of the observed data that was used to develop a model. Extrapolation is used synonymously with prediction.

This is when one or more explanatory variables can be exactly expressed as a linear combination (with various numerical coefficients) of other variables.

Assuming ill conditioning is not a problem, the estimation of a regression model for $g(t)$ might result in standard errors that are so large as to make most of the estimated coefficients unreliable (Pindyck and Rubinfeld, 1981). Even if a statistically significant regression equation for $g(t)$ could be estimated, the result may not be useful for interpolation and extrapolation. This could be attributed to any or all of the following reasons:

(a) Statistical inferences derived from least squares regression normally require that the residuals satisfy the Central Limit Theorem. If this is violated robust regression methods such as ridge regression or Maximum Likelihood Estimation can be employed. The only problem is that up to now, no paradigm has been developed that will specifically advocate a particular treatment for problems that require robust regression (Draper and Smith, 1981).

(b) The plot of the residuals against the predicted variable, $g(t)$, or explanatory variables may not be conforming to the 'horizontal band' but indicating nonlinearity not accounted for by the model or an increasing variance over time (Draper and Smith, 1981). Such a model would certainly not be reliable for predictive purposes; maybe for simulation, but only if the standard errors of the estimated coefficients of the model are small, i.e. when the fit of the model to the data is good. Possible remedies include transformation of the variables or increasing the order of the model in question.

(c) The explanatory variables that are not lagged (or delayed) may themselves have to be estimated, and this may be more difficult than estimating $g(t)$ itself (Pindyck and Rubinfeld, 1981). The standard error of the estimate of $g(t)$ with expected values of the known explanatory variables may be small (if the regression equation fits well). However, when the expected values of the explanatory variables are unknown, their estimation errors may be so large as to make the estimation for $g(t)$ unacceptable.

Let's look, for instance, at a conventional growth model that is used by many foresters for predicting stand basal area of a plantation species on known productivity sites (identified by stand predominant heights at some specific age of a forest). The Sullivan and Clutter (1972) stand basal area model (2.19) based on the Schumacher (1939) model has a structure with two unspecified parameters (C_1 and C_2) that can be fitted by means of data:

$$BA_2 = Exp\left[\frac{A_1}{A_2}\, lnBA_1 + C_1\left(1 - \frac{A_1}{A_2}\right) + C_2\left(1 - \frac{A_1}{A_2}\right)S\right] \qquad (2.19)$$

where BA_1 = initial stand basal area ($m^2\ ha^{-1}$)

BA_2 = final stand basal area ($m^2\ ha^{-1}$)

A_1 = initial stand age (years)

A_2 = final stand age (years)

S = site index (i.e. dominant height (m) at age 20)

ln = natural logarithms

C_1 and C_2 = coefficients.

Suppose there is lack of fit between data and model (2.19), and it has been verified that this lack of fit is not due to randomness or data variance but that the modelling structure is not capable of adequately explaining the observed trend. The modelling exercise then becomes one of altering the model structure.

The model (2.19) can be expressed as follows:

$$A_2 ln BA_2 = A_1 ln\, BA_1 + A_2 k \qquad (2.20)$$

where $$k = \left[C_1 \left(1 - \frac{A_1}{A_2} \right) + C_2 \left(1 - \frac{A_1}{A_2} \right) S \right]$$

From equation (2.20) it is easy to see that model (2.19) has no shape parameter other than $A_2 k$ that determines the maximum value of stand basal area. This makes model (2.19) inflexible in representing a class of models. There is also no obvious systematic means of increasing the parametric complexity of model (2.19) so that it can capture a more complex trend. An attempt to increase its parametric complexity can be difficult to formulate and time consuming; the potential of statistical and mathematical complexity is high. It may not be inconceivable to select a completely different modelling structure with all the drawbacks this entails.

It is no wonder model (2.19) could not be reliably used to model thinned stands, even after incorporating a thinning correction factor (Brack, 1985; Chikumbo, 1991; McMullan, 1978). However, Knoebel et al., (1986) overcame this thinning problem by using three models with model structure (2.19) parameterised for:

(a) before thinning;

(b) after first thinning; and

(c) after second and subsequent thinnings.

Thinning alters the shape of the basal area trend and a nonlinear function is desirable that is sensitive to changes in stand density, and is capable of mapping the corresponding changes in the basal area trend. In this dissertation such a model requirement has been met by using nonlinear dynamical models (see chapter four).

A dynamical model, in contrast, comes from a class of 'flexible' model sets (see chapter three) that can explain a variety of system behaviours without looking into their internal structures. These model sets have increasing levels of parametric complexity with corresponding capabilities of capturing increasing complex behaviour in trends. As a result if one modelling structure is unsuitable, the next level can be easily and promptly tried.

The simplest dynamical model (first order) such as the mean stand diameter model (2.21) developed by Chikumbo et al., (1992) has two parameters, *a* and *b*, that control shape and scale of the mean stand diameter trend respectively:

$$D_m(t)=aD_m(t\text{-}1) + b(1\text{-}a) \qquad (2.21)$$

where

D_m = mean stand diameter (cm)

There is thus a level of flexibility in model (2.21) that is not achievable in model (2.19).

Another popular growth function used often by foresters is the von Bertalanffy's generalised Chapman-Richards' model that represent a family of curves and is as follows:

$$y(t) = b(1\text{-}e^{-at})^c \qquad (2.22)$$

where $y(t)$ is the growth response, b, the asymptotic value of biological carrying capacity, a, the shape parameter, and c, the point of inflection in the growth curve. Model (2.22) is essentially a dynamical model in continuous-time and in discrete-time it becomes:

$$[y(t)]^{1/c} = e^{-aT}[y(t-1)]^{1/c} + b(1-e^{-aT}) \qquad (2.23)$$

where T = one time period.

Model (2.21) relates to (2.23) as follows:

$a \leftrightarrow e^{-aT}$;

$b(1-a) \leftrightarrow b(1-e^{-aT})$; and

$D_m(t) \leftrightarrow [y(t)]^{1/c}$.

However, data acquisition in forestry (or any other discipline) is in discrete-time and it follows that discrete-time model development is adequate. Ratkowsky (1983) demonstrated that the least squares estimates of the parameters a and c in model (2.22) undergo considerable variation making it hard to estimate them. This is because model (2.22) requires detailed sampled data to estimate its parameters. Such data are expensive to obtain.

This parameter instability led Ratkowsky (1983) to recommend 'reparameterisation' where one of the offending (sensitive) parameters is expressed as a function of the other parameters of the same model, without the expression containing the explanatory variables or the error term. Maximum likelihood estimation can be used for specifying the loss function or the objective function to maximise (Press et. al., 1992). Nonlinear least squares routines (Press et. al., 1992) such as the Levenberg-Marquardt method (also called Marquardt method) can be employed for the gradient search, because of their increased robustness and iterative efficiency.

Growth and yield models in forestry are mostly used for projection; forest planners want to know, with reasonable accuracy, the response of a tree crop to improved stock, cuttings, different densities etc., (Dargavel, J. per. comm., 1992). Data used for modelling are always collected from a particular site or sites and crops are well looked after which is atypical in the plantations or natural forests. Structural models developed from these data may not behave as expected when used for interpolations and extrapolations on different sites. Landsberg et al., (1985) pointed

out that, 'A model constructed from experimental data cannot be reliably extrapolated beyond its range of data ...'.

There is, however, a greater chance of minimising the problem by modelling the dynamics or behaviour of a forest plantation. This approach accounts for the external influences affecting growth, as disturbance. The disturbances may, in some cases, be separately measured, but typically are noticeable only via their effect on the growth responses. If the impulse response of a forest plantation is known, then the actual value of the disturbance, $H(q)e(t)$, can be calculated from equation (2.11) at time t. It requires less data and is mathematically simpler to deal with model disturbance, in adapting an existing dynamical model to other sites than analysing all the key variables that influence growth in those sites and then developing different growth models (see chapter six). The adaptation process is in two parts:

(a) generating a suite of curves from one growth model (developed from one site) and using minimal data points from other sites as guide points for calibration of the growth model; and

(b) setting up a recursive identification procedure for the 'guessed' model from (a), where new data, when available, can be incorporated into the model to fine tune it (see chapter six). This process can be continued until there is no appreciable changes in the parameters with further addition of new data.

Note that prediction in the long term requires that the physiology/physics of the system be understood so that forecasting can be reliably carried out within confidence intervals determined from the known behaviour of the system. In other words, a mechanistic approach is highly desirable for developing a long term prediction model. Landsberg et al., (1985) define mechanistic models of forest growth as mathematical formulations of the biological processes that occur in forests. West, et

al., (1989) point out that, because of the firmer biological basis of mechanistic models, they predict better than empirical models (black box models), the changes that occur in growth behaviour when management circumstances change.

In principle, dynamical models may be subject to some of the shortcomings discussed before, for general nonlinear regression models. However, general experience in systems engineering indicates that the dynamical models are more robust because of their structure. Indeed, a minimum number of variables is treated; there is no time (age) trajectory estimation except for the governing equations. Dynamical modelling results in a reduction in complexity and hence increased model robustness, at least intuitively. This thesis demonstrates that the intuition holds in the case of forest/silviculture predictive models.

2.5 Modelling Software: MatLab

Specialised computer languages have been developed to facilitate the task of using simulation models. Table 1 (Rimmington, 1988) lists the advantages and disadvantages of simulation programming languages. The prime advantage is the ease of implementation of the basic constructs or ideas that are used in simulation, such as integration, matrix operations, vectors and matrix manipulation etc. For example, the MatLab (MathWorks, Inc., 1984-92) statement (from the System Identification Toolbox);

y = idsim ([u e], th) ; or

y = idsim (u, th)

simulates systems specified in the theta format[9]. *th* describes an arbitrary model. *idsim* returns *y* containing the simulated output, corresponding to the input sequence *u* (one column for each input) and the noise *e*. If *e* is omitted, a noise free simulation is obtained.

In a programming language such as PASCAL or FORTRAN, the above statements would have to be replaced by a series of statements in procedures or subroutines. Having to be concerned about repetitive programming of procedures or subroutines, tends to deviate the attention away from the problem of simulation.

It is relatively straight forward to decide which simulation language to use, depending on the objectives of the simulation exercise. More than one language can be used to solve one problem although the flexibility of some languages is more constrained than others (Rimmington, 1988; Shannon, 1975). Many of these languages have been transient and only a few are well known and widely used. CSMP, DYNAMO, SIMSCRIPT, GASP, SIMULA and SIMCOM would be among the latter. There are, of course, many other simulation languages which have not been mentioned and further examples can be obtained from general simulation texts such as that of Shannon (1975).

---

[9] Theta format: This is a basic format for representing models in the system identification toolbox. It stores all relevant information about the model structure used, including the values of the estimated parameters, the estimated covariances of the parameters, and the estimated innovations variance (see appendix I) and so on (Ljung, 1991).

Table 1: The advantages and disadvantages of using simulation programming languages compared with using more common programming languages (Rimmington, 1988)

---Advantages:

1. Less programming time.
2. Allows for direct expression of simulation constructs.
3. Usually have many inbuilt functions and subroutines which are required in simulations e.g. interpolation, integration.
4. Automatic code generation (defaults set-up in programs).
5. Automatic generation of certain data e.g. random numbers for different probability distributions.
6. Automatic printing and graphical display of data.
7. Debugging in simulation terminology.

Disadvantages:

1. Less familiar to most experienced programmers.
2. Less available for computers than common languages.
3. Less flexible input/output formats.
4. Require more computer memory and time to run (Note that technology is becoming cheaper and hence powerful computers are becoming affordable)
5. The user may not fully understand the underlying mathematical or computational techniques used by the language resulting in incorrect use or misinterpretation of results.

2.5.1 MatLab Software

All the system identification problems in this dissertation are computed and analysed using MatLab.

MatLab stands for *matrix laboratory*. It is a high-performance interactive software package adapted for scientific and engineering numeric computation. MatLab integrates numerical analysis, matrix computation, signal processing and graphics in an easy-to-use environment where problems and solutions are expressed just as they are written mathematically, without traditional programming. The basic data element is a matrix that does not require dimensioning.

Entirely written in the C language, MatLab is a complete integrated system, including graphics, programmable macros, IEEE arithmetic, a fast interpreter

and many analytical commands. Optional toolboxes are available to extend MatLab's functionality, providing additional application-specific capabilities. It is for this reason that the software package has been chosen for dynamical modelling analysis. The System Identification Toolbox (a collection of M-files[10]) by Prof. Lennart Ljung adds commands for parametric and non-parametric modelling and system identification. It is designed for estimating models of a system based upon input/output data, or on time series. The model structures, such as ARMA, ARMAX, AR, ARX etc., are outlined in chapter three.

The System Identification Toolbox is matched directly to the textbook by Ljung (1987) which is a useful complement for learning the topic.

10 Script files and function files are collectively called M-files in MatLab.

Endnote 2A: Detrending and Stabilisation of Variance

Detrending in a time-series data set involves changing *non-stationary* data to *stationary* data. Stationarity is defined by a constant mean, a constant variance and an autocorrelation between values of the process at two time periods (Vandaele, 1975). Autocorrelation measures the correlation between an observation at time t, $u(t)$, and an observation at time s, $u(s)$. Thus to estimate the mean, variance and autocorrelations of a stochastic process, some restrictions should be imposed on the data:

(a) If there is no trend in the data, assume the mean is constant for each time period;

(b) Assume the variance of the process is constant. This expresses the degree of variation around the assumed constant mean level; and

(c) The autocorrelation between values of the process at two time periods, say t and s, must depend only on the distance between these two points and not on the time period itself (assuming $t \neq s$).

The first step in any series analysis is to plot the available observations against time to reveal any qualitative features such as trend, seasonality, discontinuities and outliers. If the data series shows non-stationarity characteristics then some treatments can be followed until the above restrictions are met.

Stabilisation of the Variance

The basic idea is to transform the data such that an originally curved plot is straighter and at the same time make the variance constant over the whole series. Two of the frequently used transformations are the logarithmic transformation and the square root transformation.

Removal of Trend

Trend can be described as any long-term, systematic and monotonic movement relative to a time sequence data set. A regression model is a common method of removing a trend. Such a regression model can be written as

$$y(t) = b_0 + b_1 x + e(t) \tag{2A.1}$$

if the trend is assumed to be linear, or as

$$y(t) = b_0 + b_1 x + b_2 x^2 + e(t) \tag{2A.2}$$

if the trend is presumed to be a quadratic polynomial.

In the above models, an assumption is made that the trend is fixed and deterministic.

Other methods such as differencing can be employed to remove a trend due to bias. Detail on differencing can be found in books by Box and Jenkins (1971) and Vandaele (1975). Differencing seems quite useful when modelling a stochastic trend (Vandaele, 1975).

Care, however, should always be taken when detrending as this depends on the type of data. Consider, for example, a first order model with a constant input of ones. MatLab is used to identify a unit step response for the first order discrete time model with a pole at 0.8, i.e.

$$y(t) = 0.8y(t - 1) + 0.2u(t) \quad t= 1 .. 100, \tag{2A.3}$$

before detrending.

Detrending attempts to remove the input. The average that is subtracted will be wrong due to initial conditions and the fact that only a finite time of observations exists. The detrending procedure gives a different model:

$$y(t) = 0.7998y(t- 1) \tag{2A.4}$$

The models (2A.3) and (2A.4) are different and thus detrending is important in certain circumstances: when a substantial amount of input data points are unknown, which vary a lot and have some deterministic non-zero average, then detrending can be applied to get the accurate system output, otherwise the offset for

transients in the system will be misinterpreted and hence the wrong time constants simulated.

CHAPTER 3

Models of Linear Time-Invariant Systems

System identification results in the derivation of specific model structures. This chapter introduces the basic properties of these model structures in discrete-time. A discrete-time response represents a phenomenon of an independent variable takes distinct values. Its system describes the process that results in the transformation of the signal (Oppenheim and Willsky, 1983). Linearity and time-invariance play a fundamental role in system analysis because linear time-invariant (LTI) systems can represent many physical processes.

A linear system in discrete time is one that possesses the important property of *superposition*. If an input consists of the weighted sum of several signals, then the output is simply the superposition, i.e. the weighted sum of the responses of the system to each of those signals (Luenberger, 1979; Oppenheim and Willsky, 1983). Let $y_1(t)$ be the response of a discrete-time system to $u_1(t)$ and let $y_2(t)$ be the output corresponding to the input $u_2(t)$. The system is linear if:

(a) the response to $u_1(t) + u_2(t)$ is $y_1(t) + y_2(t)$; and

(b) the response to $au_1(t)$ is $ay_1(t)$, where a is any real constant.

A system is *time-invariant* if only a time shift in the input signal causes a time shift in the output signal (Jenkins and Watts, 1968; Oppenheim and Willsky, 1983). Specifically, if $y(t)$ is the output of a discrete-time time-invariant system when $u(t)$ is the input, then $y(t - \tau)$ is the observed response that results from an input of $u(t - \tau)$.

Typically, influences other than u will affect y. A model that can be related to measurement data must take into account not only the dynamic relationship associating u and y but also the noise affecting the system. Such joint models (as in

equation (2.11)) are obtained by combining a deterministic transfer function with a stochastic noise model.

A physical interpretation of time invariance in a forestry context, would be a response of growth models over the second and subsequent rotations. Given forest growth models for a particular site, it would be expected that the growth trends remain the same in the second and subsequent rotations provided the site conditions stay the same. The growth models would therefore depend on the site conditions being invariant although in reality, even with human intervention, that is hard if not impossible to guarantee.

3.1 Model Structures

A complete LTI model is represented by equation (2.11). A particular model thus corresponds to specifications of the three functions, G, H and f_e (probability density function of e). In most cases it is impractical to enumerate the infinite sequences $g(k)$, $h(k)$ together with the function $f_e(x)$. To overcome this problem, structures that permit specification of G and H in terms of a finite number of numerical values, are used: rational transfer functions and finite state space descriptions are examples (Ljung, 1987). Most often the probability density function, f_e, is not specified as a function, but described in terms of a few numerical characteristics, typically the first and second moments:

$$E\, e(t) = \int x f_e(x)\, dx = 0$$

and (3.1)

$$E\, e^2(t) = \int x^2 f_e(x)\, dx = \lambda$$

It is common to assume that $e(t)$ is Gaussian (Ljung, 1987), in which case the probability density function is entirely specified by (3.1).

The specification of equation (2.11) in terms of finite number of numerical values or coefficients, is in most cases, left to estimation procedures. The coefficients, therefore enter the model (2.11) as parameters to be determined (Ljung, 1987). The notation used to denote such parameters will be the vector θ. The model (2.11) thus becomes:

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t) \tag{3.2a}$$

$$(f_e(x,\theta) \text{ is the pdf of } e(t);\ e(t) \text{ is white noise}) \tag{3.2b}$$

The parameter vector θ ranges over a subset of R^d, where d is the dimension of θ. The model (3.5) is no longer a single function but a set of functions. Estimation of θ that is suitable for the modelling requirements will rely on the estimation procedure and the nature of the available data. Using (2.14), the one-step-ahead prediction for (3.2) can be denoted by:

$$\hat{y}(t|\theta) = H^{-1}(q,\theta)G(q,\theta)u(t) + [1 - H^{-1}(q,\theta)]y(t) \tag{3.3}$$

This predictor form does not depend on $f_e(x, \theta)$. Models that specify (3.2a) only, are referred to as *predictor models* and those that specify (3.2a) and (3.2b), as *probabilistic models*. The parameterised set of models such as (3.3) are called model structures (Ljung, 1987).

3.2 Black-box models

G and H are represented as rational functions and the parameters become the numerator and denominator coefficients.

Equation Error Model structure

Suppose an input/output relationship is represented by a linear difference equation:

$$y(t) + a_1 y(t-1) + \ldots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \ldots + b_{n_b} u(t-n_b) + e(t) \tag{3.4}$$

Since the white noise term enters as a direct error in the equation, the model (3.4) is often called an *equation error model* (structure). θ (adjustable parameters) is defined as:

$$\theta = [a_1\ a_2\ \ldots\ a_{n_a}\ b_1\ \ldots\ b_{n_b}]^T \tag{3.5}$$

If $A(q)$ and $B(q)$ are introduced such that:

$$A(q) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-na} \tag{3.6}$$

and

$$B(q) = b_1 q^{-1} + \ldots + b_{n_b} q^{-nb} \tag{3.7}$$

then (3.7) will correspond to (3.5) with:

$$G(q, \theta) = \frac{B(q)}{A(q)} \qquad H(q, \theta) = \frac{1}{A(q)} \tag{3.8}$$

The model (3.4) is also called an ARX model (see Endnote 3A) where AR refers to the *A*uto*R*egressive part, $A(q)y(t)$, and X to the extra input, $B(q)u(t)$, called an exogenous variable in econometrics (Pindyck and Rubinfeld, 1976). In special cases where $n_a = 0$, $y(t)$ is modelled as a finite impulse response (Ljung, 1987). The input affects the output only over a finite time given by n_b.

ARMAX Model Structure

The basic disadvantage with the simple model (3.4) is the lack of adequate freedom in describing the properties of the disturbance term. This flexibility can be achieved by introducing the equation error as a *moving average* of white noise (see Endnote 3B). This gives the model:

$$y(t) + a_1 y(t - 1) + \ldots + a_{n_a} y(t - n_a)$$

$$= b_1 u(t - 1) + \ldots + b_{n_b} u(t - n_{b)} + e(t) + c_1 e(t - 1) + \ldots + c_{n_c} e(t - n_c) \tag{3.9}$$

With

$$C(q) = 1 + c_1 q^{-1} + \ldots\ c_{n_c} q^{-n_c} \tag{3.10}$$

the model can be written as:

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \tag{3.11}$$

and clearly corresponds to (3.2) with:

$$G(q, \theta) = \frac{B(q)}{A(q)}, \qquad H(q, \theta) = \frac{C(q)}{A(q)} \tag{3.12}$$

where

$$\theta = [a_1 \ldots a_{n_a} \;\; b_1 \ldots b_{n_b} \;\; c_1 \ldots c_{n_c}]^T \tag{3.13}$$

In view of the moving average (MA) part, $C(q)e(t)$, the model (3.11) is called ARMAX. ARMAX is a standard tool in control and econometrics for both system description and control design (Ljung, 1987). A version with the enforced integration in the system description is called the ARIMA(X) model, with or without the X-variable u, which is useful to describe systems with slow disturbances (Box and Jenkins, 1971; Vandaele, 1975). It is obtained by replacing $y(t)$ in (3.11) by $\Delta y(t) = y(t) - y(t-1)$.

The predictor for (3.11) is obtained by inserting (3.12) into (3.13):

$$\hat{y}(t|\theta) = \frac{B(q)}{C(q)}\, u(t) + \left[1 - \frac{A(q)}{C(q)}\right] y(t) \tag{3.14}$$

or

$$C(q)\, \hat{y}(t \mid q) = B(q)u(t) + [C(q) - A(q)]\, y(t) \tag{3.15}$$

This means that the prediction is obtained by filtering u and y through a filter with denominator dynamics determined by $C(q)$.

3.3 State -Space Models

Perhaps the most crucial notion in dynamics is the concept of state and other than having an important philosophical value, it is of great practical relevance in fields of application. Though the input/output framework is the natural

representation for system identification, the input/state/output framework is conducive for control design purposes and hence the need to discuss the state space ideas.

State variables express the memory of a system. In other words, states are summaries of the information contained in the exogenous signal that is transmitted by the dynamics generated by a time series data. The state vector, for example, of a deterministic dynamical system is a minimal collection of information necessary to determine uniquely the future evolution of a dynamical system, given future time paths of all relevant exogenous variables affecting the system, including decision or control variables.

The notion of state and the problem of state representation are easily introduced on a level of generality of Σ as in section 2.1.1. A state space system is defined as a quadruple $\Sigma_i = \{T, W, X, \beta_i\}$ with $T \subset R$ the time set, W the (external) response variable, X the state space, and $\beta_i(W \times X)^T$ the internal behaviour (Willems, 1986a). It is assumed that β_i satisfies the axiom of state which states:

$$\{(w_k, x_k) \in \beta_i, k = 1, 2,\ t_0 \in T, x_1(t_0) = x_2(t_0)\} \Rightarrow \{(w_1, x_1) \underset{t_0}{\wedge} (w_2, x_2) \in \beta\}$$

where

$\underset{t_0}{\wedge}$ is the concatenation product at t_0.

The axiom of state implies that any path leading to a particular state will be compatible with any other path emanating from that same state. In other words, in a set theoretic sense, the past and future are *conditionally independent* given the present state, i.e. the present state splits the past and the future behaviour of the system.

Let $\Sigma_i = \{T, W, X, \beta_i\}$ be a state space system. Then $\Sigma := \{T, W, \beta\}$ with $\beta := \{w : T \rightarrow X \mid \exists x : T \rightarrow X$ such that $(w, x) \in \beta_i\}$ is called the dynamical system induced by Σ_i and β is called the external behaviour of Σ_i. This can be denoted as

$\Sigma_i \Rightarrow \Sigma$ and $\beta_i \Rightarrow \beta$. If Σ is given and Σ_i is such that $\Sigma_i \Rightarrow \Sigma$ then Σ_i is a state space representation (or realisation) of Σ.

One of the main problems treated in mathematical systems theory is the one of representing a given dynamical system as the external behaviour of a system in state form. The problem is to find an irreducible representation which has a convenient form. The basic problem treated is as follows: given an (external) dynamical system Σ, find a state space system Σ such that $\Sigma_i \Rightarrow \Sigma$. This problem has been studied by Willems (1979) in a set theoretic context.

The purpose of the state variable model is to develop a representation that preserves the input/output relationship (of the transfer function) and is expressed in n first-order equations for the n^{th} order system. The advantage of the n first-order equations is that, in addition to the input/output characteristics, the internal characteristics of the system are represented. Important reasons for developing the state model (in chapter five) are as follows:

(a) Design procedures that result in the 'best' control system are almost all based on state variable models. By 'best' it is meant that the system has been designed in such a way as to optimise a mathematical function that expresses the design criteria.

(b) Computer-aided analysis and design of state models are performed more easily on digital computers for higher-order systems, while the transfer function approach may fail for these systems;

(c) In state variable design procedures, more information (internal variables) is fed back into the system, hence a more

complete control of the system is performed than is possible with a transfer function approach;

A common way of describing linear systems using the state-space form is as follows:

$$x(t+1) = Ax(t) + Bu(t) \tag{3.16a}$$

$$y(t) = Cx(t) + Du(t) + v(t) \tag{3.16b}$$

where

(3.16a) is called the state transition equation and (3.16b) the output or observation equation;

$x(t)$ = state vector: $(n \times 1)$ vector of the states of an n^{th}-order system;

A = $(n \times n)$ matrix called the system matrix (or transition matrix);

B = $(n \times r)$ matrix called the input matrix;

$u(t)$ = input vector: $(r \times 1)$ vector composed of the system input functions;

$y(t)$ = output vector: $(p \times 1)$ vector composed of the defined outputs;

C = $(p \times n)$ matrix called the output matrix; and

D = $(p \times r)$ matrix to represent direct coupling between input and output.

State-space models are theoretically equivalent to the transfer function type because models in one representation can be transformed into the other (Aoki, 1987). Judged on the grounds of numerical stability, sensitivity with respect to small specification errors, statistical properties of parameter estimates or simply ease of handling for vector-valued or non-stationary time series, the two types of modes are not equivalent. For control purposes there is a preference for state space models.

If a system identification technique is used to obtain the system model, only a transfer function may be available to describe the system. For this and other reasons, it is advantageous to have a method available to obtain a state-space model

directly from a transfer function. For example, suppose an LTI discrete-time system is represented by a second-order equation[1], {*ARX(2 2 0)*},

$$y(t) = a_1 y(t - 1) + a_2 y(t - 2) + b_1 u(t - 1) + b_2 (t - 2) \tag{3.17}$$

The z-transform of equation (3.17) yields the transfer function:

$$\frac{Y(z)}{U(z)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} = \frac{b_1 z + b_2}{z^2 - a_1 - a_2} \tag{3.18}$$

A state-space model of the system described by (3.18) can be developed by assigning a state variable to the output of each unit delay, $x_1(k)$ and $x_2(k)$ respectively as in the simulation diagram in Figure 3.1. The input to the first delay then becomes $x_1(k + 1)$ and the input to the second one, $x_2(k + 1)$. Thus

$$\begin{aligned} x_1(k + 1) &= x_2(k) \\ x_2(k + 1) &= a_2 x_1(k) + a_1 x_2(k) + u(k) \\ y(k) &= b_2 x_1(k) + b_1 x_2(k) \end{aligned} \tag{3.19}$$

These equations can be expressed in a vector form as,

$$\begin{bmatrix} x_1(k + 1) \\ x_2(k + 1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_2 & a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [b_2 \quad b_1] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{3.20}$$

The above form is known as the controller canonical form. Note that the choice of state variables is by no means unique.

[1] See Endnote 3C for explanation of first or second order dynamical models.

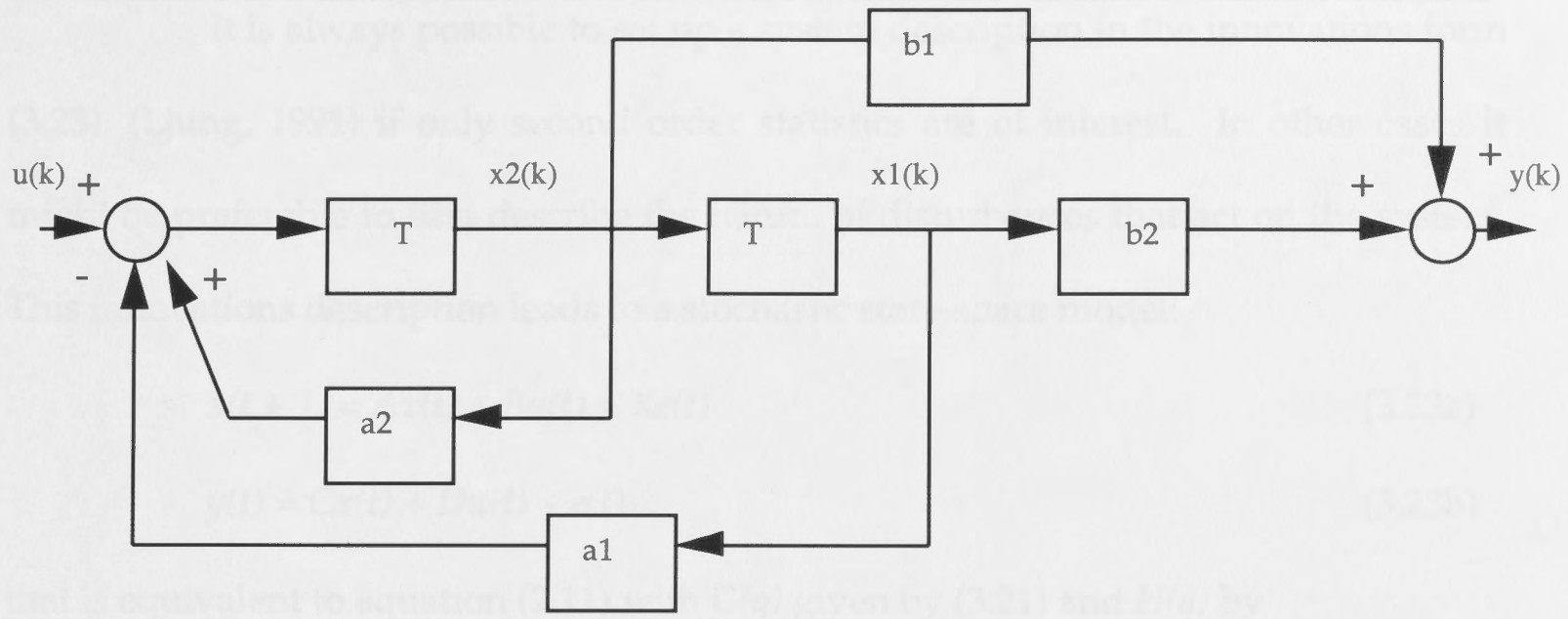


Figure 3.1 : Simulation diagram for equation (3.17)

It can now be seen that transfer function form (3.16) corresponds to equation (2.2) and the general function of (3.21) links the two:

$$G(q) = C(qI_{nx} - A)^{-1}B + D \qquad (3.21)$$

where I_{nx} is the nx x nx identity matrix and nx is the dimension of x (Ljung, 1991). Thus (3.16) can be viewed as one way of parametrising a transfer function: via (3.21), where $G(q)$ becomes a function of the elements of the matrices A, B, C and D. Given $x(t)$ and $u(t)$, the output equation yields the output $y(t)$. Usually the matrix D is zero, since in physical systems, dynamics appear in all the paths between the inputs and the outputs. A non-zero value of D indicates at least one direct path between the inputs and the outputs, in which the path transfer function can be modelled as a pure gain.

To further describe the character of the noise term $v(t)$ in (3.21) a more flexible form of the state-space model can be used:

$$x(t + 1) = Ax(t) + Bu(t) + w(t) \qquad (3.22a)$$

$$y(t) = Cx(t) + Du(t) + e(t) \qquad (3.22b)$$

where

$w(t)$ and $e(t)$ are stochastic processes with certain covariance properties.

It is always possible to set up a system description in the innovations form (3.23), (Ljung, 1991) if only second order statistics are of interest. In other cases it might be preferable to first describe the nature of disturbances that act on the system. This innovations description leads to a stochastic state-space model:

$$x(t+1) = Ax(t) + Bu(t) + Ke(t) \tag{3.23a}$$

$$y(t) = Cx(t) + Du(t) + e(t) \tag{3.23b}$$

that is equivalent to equation (2.11) with $G(q)$ given by (3.21) and $H(q)$ by

$$H(q) = C(qI_{nx} - A)^{-1}K + I_{ny} \tag{3.24}$$

where ny is the dimension of $y(t)$ and $e(t)$.

In stationarity and from an input/output point of view, (3.23) is equivalent to (3.22) if the matrix K is chosen as the steady-state *Kalman gain* (Ljung, 1991).

Endnote 3A: AR process

A stationary time series is said to be governed by a first-order autoregressive (AR) process if the current value of the time series $y(t)$ can be expressed as a linear function of the previous value of the series and a random shock $e(t)$. The AR process is constructed from (conditional) multivariate regressions. The vector $y(t-1)$ constitutes the independent variables and the vector $y(t)$ constitutes the dependent variables in ordinary regression. To a large extent the statistical methods for AR models are regression or least-squares procedures (Anderson, 1978). The process can be written as

$$y(t) = fy(t - 1) + e(t) \tag{3A.1}$$

where f, is the AR parameter that describes the effect of a unit change in $y(t - 1)$ for predicting $y(t)$, and needs to be estimated (Vandaele, 1975). The random shocks $e(t)$ also known as error or white noise series, are assumed to be normally and independently distributed with mean zero, constant variance δ_e^2 and independent of $y(t - 1)$; i.e.

$$E\, e(t) = 0 \tag{3A.2}$$

$$E\, e(t)\, e(s) = \begin{cases} \delta_e^2 & \text{if } t = s \\ 0 & \text{if } t \neq s \end{cases} \tag{3A.2}$$

$$E(e(t)\, y(t - 1)) = 0 \tag{3A.4}$$

If normality is assumed, $e(t)$ is sometimes defined as Gaussian white noise. Alternatively (3A.1) could be presented as

$$(y(t) - \mu) = f_1(y(t - 1) - \mu) + e(t) \tag{3A.5}$$

or

$$z(t) = (1 - \phi_1)\mu + \phi_1 z(t - 1) + e(t) \tag{3A.6}$$

where μ is the mean of $z(t)$ and $z(t)$ represents the observed data. (See model (2.21)).

The only formal difference between (3A.1) and (3A.6) is the inclusion of the intercept $(1 - \phi_1)\mu$.

The process described by equation (3A.1) is called an *autoregressive process* of order 1, AR(1) (Vandaele, 1975). The order of the process corresponds to the number of parameters that need to be estimated. In the case of an autoregressive process, that number corresponds to the number of lagged y's included in the model. The variance of the AR(1) process has to be non-negative (by definition all variances are non-negative because the expectation of the square of a certain quantity cannot be negative) and it is, therefore, necessary that $|\phi_1|$ be less than or equal to 1. Vandaele (1975) gives a mathematical proof that qualifies the above statement.

Properties of an AR process

(a) Autocovariances and autocorrelations

Note that the ordinary correlation is closely related to the bivariate normal distribution. With more than two variables, an extension of this distribution called the *multivariate normal distribution* forms the basic model in correlation studies (Snedecor and Cochran, 1980). A property of the multivariate normal model is that any variable has a linear regression on the variables (or on any subset of the other variables) with deviations that are normally distributed. A brief description of correlations is therefore vital in understanding the AR process.

The covariances between 2 random variables is defined as

$$Co(x, y) = E\,[(x - Ex)(y - Ey)] \qquad (3A.7)$$

where Ex and Ey are the means of random variables x and y, respectively. The autocovariance of $y(t)$ at lag 1, denoted by λ_1, is a covariance between $y(t)$ and $y(t - 1)$ and is therefore defined by

$$\lambda_1 \equiv Cov(y(t), y(t - 1)) = E(y(t)\, y(t - 1)) \qquad (3A.8)$$

The autocovariance solely depends on the lag between $y(t)$ and $y(t - 1)$. The variance of $y(t)$ can be defined as $cov(y(t), y(t))$ and we can therefore denote the variance of $y(t)$ as λ_0.

Substituting (2A.1) for $y(t)$ in (2A.8) and using the assumption $E\, e(t) = 0$,

$$\lambda_1 = \phi_1 Var(y(t - 1)) + Cov(e(t), y(t - 1)) \tag{3A.9}$$

Since $e(t)$ is independent of $y(t - 1)$, the last term on the right hand side of (3A.9) is zero hence

$$\lambda_1 = \phi_1 Var(y(t - 1)) \tag{3A.10}$$

Because stationarity implies that $\lambda_0 \equiv Var(y(t)) = Var(y(t - 1))$, (3A.10) becomes

$$\lambda_1 = \phi_1 \lambda_0 \tag{2A.11}$$

The autocovariance at lag 2, λ_2, is defined as $cov(y(t), y(t - 2))$. Substituting (3A.1) for $y(t)$ in (3A.8) λ_2 becomes

$$\lambda_2 = f_1 Cov(y(t - 1), y(t - 2)) + Cov(e(t), y(t - 2)) \tag{3A.12}$$

Since the lag between $y(t - 1)$ and $y(t - 2)$ is one period, $cov(y(t - 1)(y - 2))$ is simply λ_1. (3A.12), therefore, reduces to

$$\lambda_2 = \phi_1 \lambda_1 = \phi_1^2 \lambda_0 \tag{3A.13}$$

It can thus be shown that for $k > 0$,

$$\lambda_k = \phi_1 \lambda_{k-1} \tag{3A.14}$$

or

$$\lambda_k = \phi_1^k \lambda_0 \tag{3A.15}$$

The autocorrelation at lag k, ρ_k is defined as the ratio of the autocovariance at lag k to the autocovariance at lag zero $\{Var(y(t))\}$ and is therefore a *scaled* autocovariance. For AR(1), the autocorrelations are obtained as

$$\rho_k = \frac{\lambda_k}{\lambda_0} = \phi_1^k \qquad k > 0 \qquad \text{(3A.16)}$$

Equation (3A.1) can be broadened to include more lagged variables. For example if events, two periods ago, had an effect on what is happening today, (3A.1) could be extended to include $y(t - 2)$, i.e.

$$y(t) = \phi_1 y(t - 1) + \phi_2 y(t - 2) + e(t) \qquad \text{(3A.17)}$$

where ϕ_1 and ϕ_2 are autoregressive parameters to be estimated. If (3A.17) is expressed in terms of the observed data, the constant added to the equation becomes $\delta = (1 - \phi_1 - \phi_2)\mu$. In general a p^{th} order autoregressive model, AR(p) is

$$y(t) = \phi_1 y(t-1) + \phi_2 y(t-2) + \ldots + \phi_p y(t-p) + e(t) \qquad \text{(3A.18)}$$

(b) Memory function

An AR process has a long memory. Suppose equation (3A.1) is written in terms of past errors by successively eliminating the lagged $y(t)$'s, i.e. substitute

$$y(t-1) = \phi_1 y(t - 2) + e(t - 1) \qquad \text{(3A.19)}$$

into (3A.1) to get

$$y(t) = \phi_1^2 y(t - 2) + e(t) + \phi_1 e(t - 1) \qquad \text{(3A.20)}$$

Then substitute

$y(t - 2) = \phi_1 y(t - 3) + e(t - 2)$ into equation (3A.20) to obtain

$$y(t) = \phi_1^3 y(t - 3) + e(t) + \phi_1 e(t - 1) + \phi_1^2 e(t - 2) \qquad \text{(3A.21)}$$

and so on until an error-shock form,

$$y(t) = e(t) + \phi_1 e(t - 1) + \phi_1^2 e(t - 2) + \phi_1^3 e(t - 3) + \ldots \qquad \text{(3A.22)}$$

is obtained (Vandaele, 1975).

Thus the AR(1) model is rewritten as a sum of the current error and an infinite number of past error terms. An AR(1) process is thus said to have an infinite memory. If the process is stationary, i.e. $|\phi_1| < 1$, the effect of shock gradually dissipates. This is not the case when the process is non-stationary.

A plot of the memory coefficients (a correlogram) as a function of the lag k, $k \geq 0$, is called the *memory function* of the process.

Endnote 3B: MA process

An AR(1) model as in (3A.1) can be modified to include past errors so as to improve on the time series representation of the data:

$$y(t) = \phi_1 y(t - 1) + e(t) - \theta_1 e(t - 1) \qquad (3B.1)$$

where $e(t - 1)$ is the error at period $t - 1$ and θ_1 is called the *moving average parameter* which describes the effect of the past error on $y(t)$ and which needs to be estimated. It is customary to write a negative sign in front of the parameter θ_1 (Vandaele, 1975). Model (3B.1) has the form of a multiple regression model with two independent variables $y(t - 1)$ and $e(t - 1)$. This analogy is imperfect because the lagged error term is not observed and as such cannot be used as a regressor.

A special model can be obtained from (3B.1) by omitting the lagged variable $y(t - 1)$. This is called a *moving average* model of order one, (MA(1)) and expresses the current value of the series $y(t)$ as a linear function of the current and previous error shocks, $e(t)$ and $e(t - 1)$. Mathematically, a first-order moving average model is expressed as

$$y(t) = e(t) - \phi_1 e(t - 1) \qquad (3B.2)$$

As the AR process, the random shocks in an MA process are assumed to be normally and independently distributed with mean zero and constant variance δ_e^2 ; i.e. they satisfy equations (3A.2) and (3A.3).

Properties of an MA process

(a) Autocovariances and autocorrelations

Vandaele (1975) shows that

$$\lambda_0 = Var(y(t)) = (1 - \theta_1^2)\delta_e^2 \qquad (3B.3)$$

and

$$\lambda_1 = Co(y(t), y(t-1)) = -\theta_1 \delta_e^2 \tag{3B.4}$$

Similarly the autocovariance at lag 2, λ_2, is obtained as

$$\lambda_2 = Co(y(t), y(t-2)) = E[(e(t) - \theta e(t-1))(e(t-2) - \theta_1 e(t-3))]$$

$$= 0 \tag{3B.5}$$

Equation (3B.5) is obtained by repeated use of equation (3A.3), $E(e(t)\, e(s)) = 0$ for $t \neq s$. Similarly it can be shown that $\lambda_k = 0$ for $k \geq 3$. Therefore, the autocorrelations defined as $\rho_k = \frac{\lambda_k}{\lambda_0}$, are

$$\rho_k = -\theta_k/(1 - \theta_1^2);$$

$$\rho_k = 0, \qquad k \geq 3. \tag{3B.6}$$

It is clear to note that the autocorrelations for the AR(1) process, specified in (3A.16), die out gradually, and those for the MA(1) process die abruptly; only ρ is not zero. By definition, the autocorrelation ρ_0 is always equal to 1, regardless of the process being analysed (Vandaele, 1975).

(b) Memory function

From equation (3B.2) it follows directly that for an MA process, a shock at time t, $e(t)$ will influence the observations at time t and $t + 1$, but will have no effect beyond $t + 1$. At time t the system will feel the full impact of the shock, and at time $t + 1$, its effect will be proportional to θ_1 (Vandaele, 1975). Therefore the memory of an MA(1) process only lasts for one period.

Endnote 3C: First- and second-order systems

First- and second-order systems can be thought of as building blocks of higher order systems and their understanding would give clear insight of the general behaviour of LTI systems.

The first order LTI system in discrete-time is described by the following difference equation:

$$y(t) = ay(t-1) + b(1-a) \qquad (3C.1)$$

The parameter a is related to the *time constant* that controls the rate at which the first-order system responds. $0 < a < 1$ corresponds to stable systems. The impulse response decays sharply when $|a|$ is small and for $|a|$ nearer to one the response is slower. Figure 3A.1 shows a suite of curves of the LTI system (3C.1) where $0 < a < 1$ and b (the scale parameter) = 1. When $-1 < a < 0$ the step response exhibits both overshoot of its final value and oscillatory behaviour.

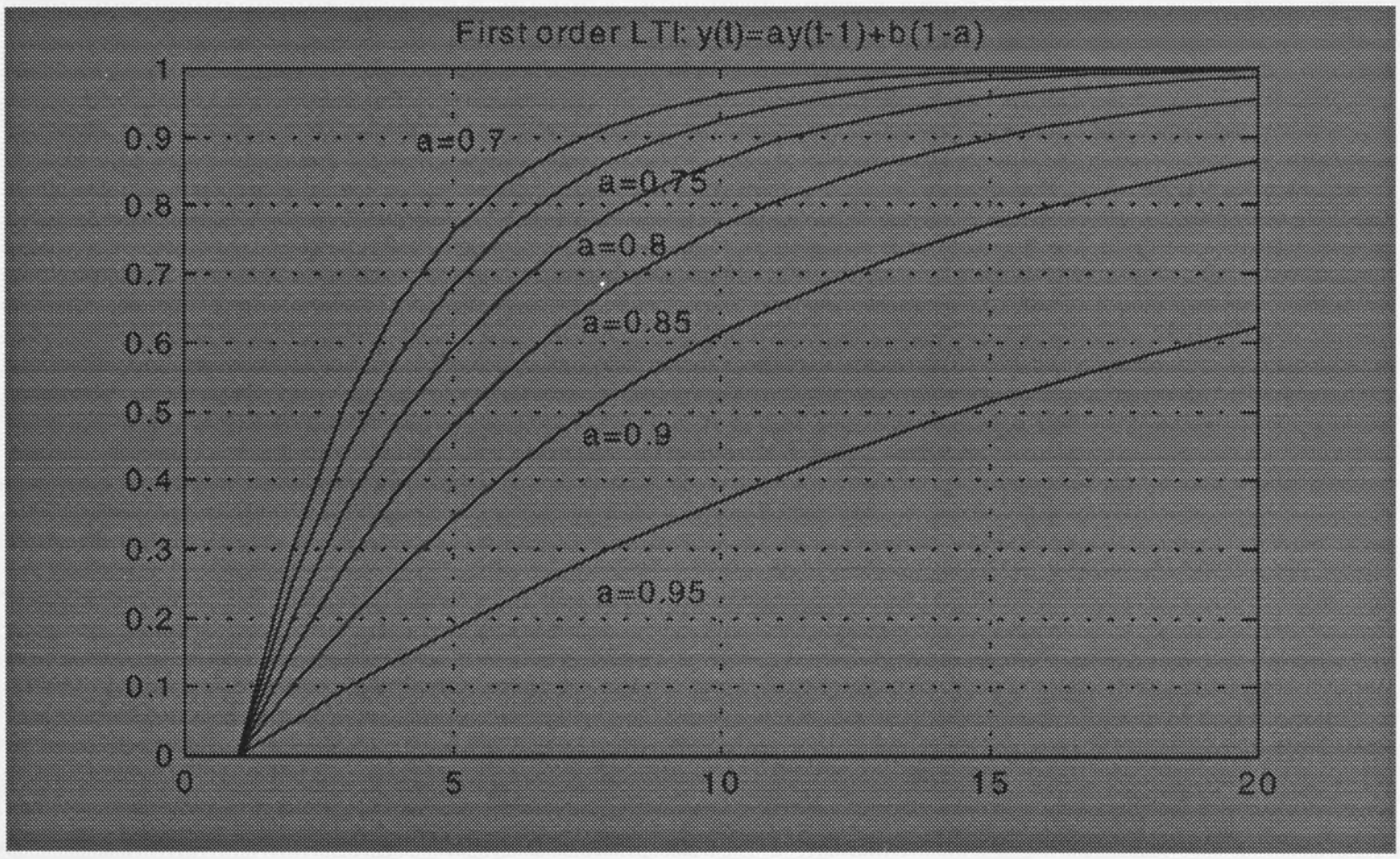


Figure 3C.1: A suite of curves of an LTI first-order system where $0 < a < 1$; $b = 1$; $y(1) = 0$; and $t = 1, 2, \ldots$.

A second-order system in discrete-time with may be represented as follows:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + b(1 - a_1 - a_2) \qquad (3C.2)$$

In order to describe a stable system the parameters a_1 and a_2 are such that $0 < a_1 < 2$ and $-\frac{a_1^2}{4} \le a_2 < 1$ (see Figure 3C.2). For non-oscillatory behaviour, $-\frac{a_1^2}{4} \le a_2 < 0$. Three cases of a second-order system can be identified which are, *underdamped* ($-4a_2 > a_1^2$), *overdamped* ($a_1^2 > -4a_2$) and *critically damped* ($a_1^2 = -4a_2$). The latter case has the potential for a wider application in forest growth modelling. Figure 3C.3 shows a series of curves of a critically damped second-order system with varying a_1 and a_2 parameters.

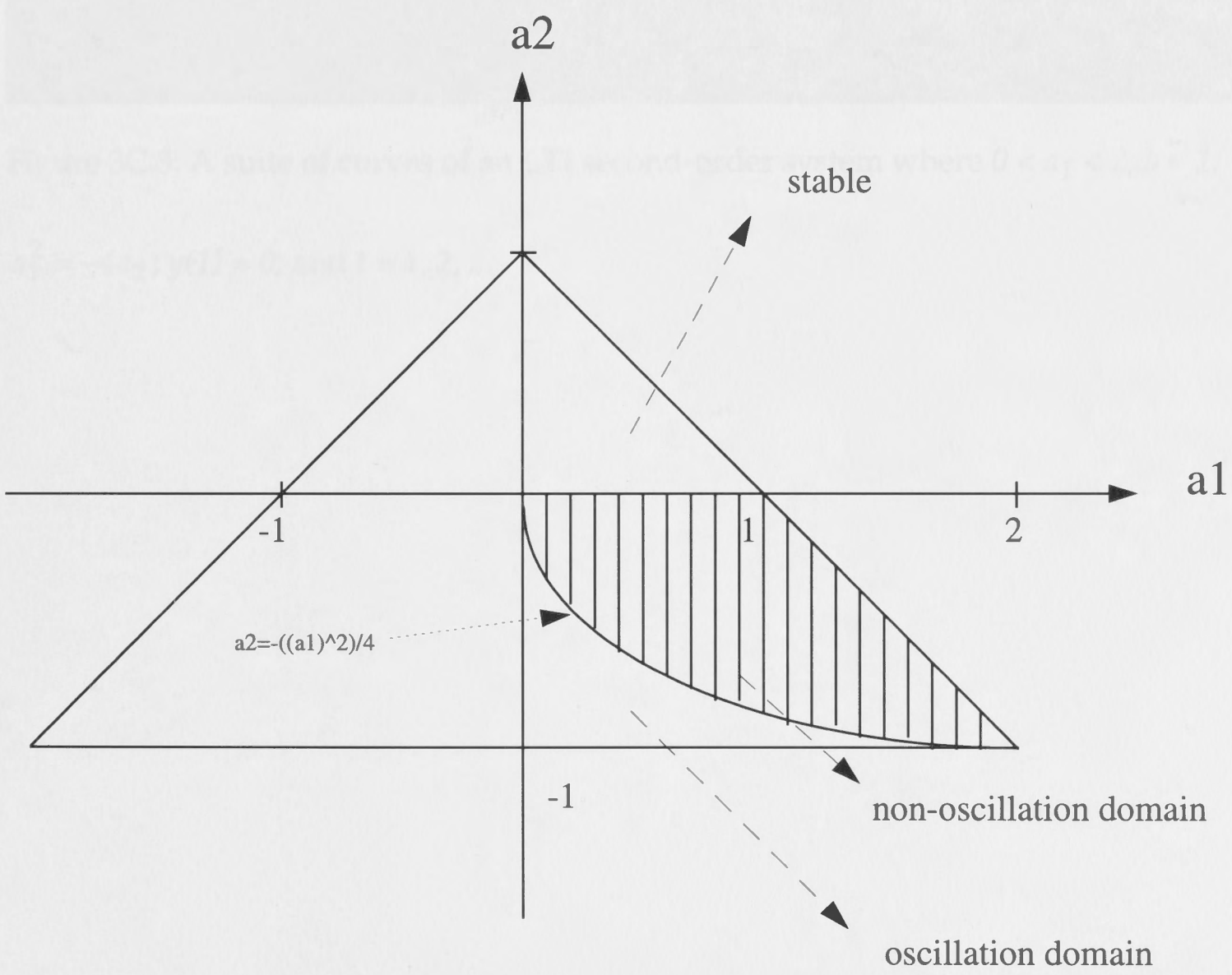


Figure 3C.2: Relationship of the time constants, a_1 and a_2, for a second-order system.

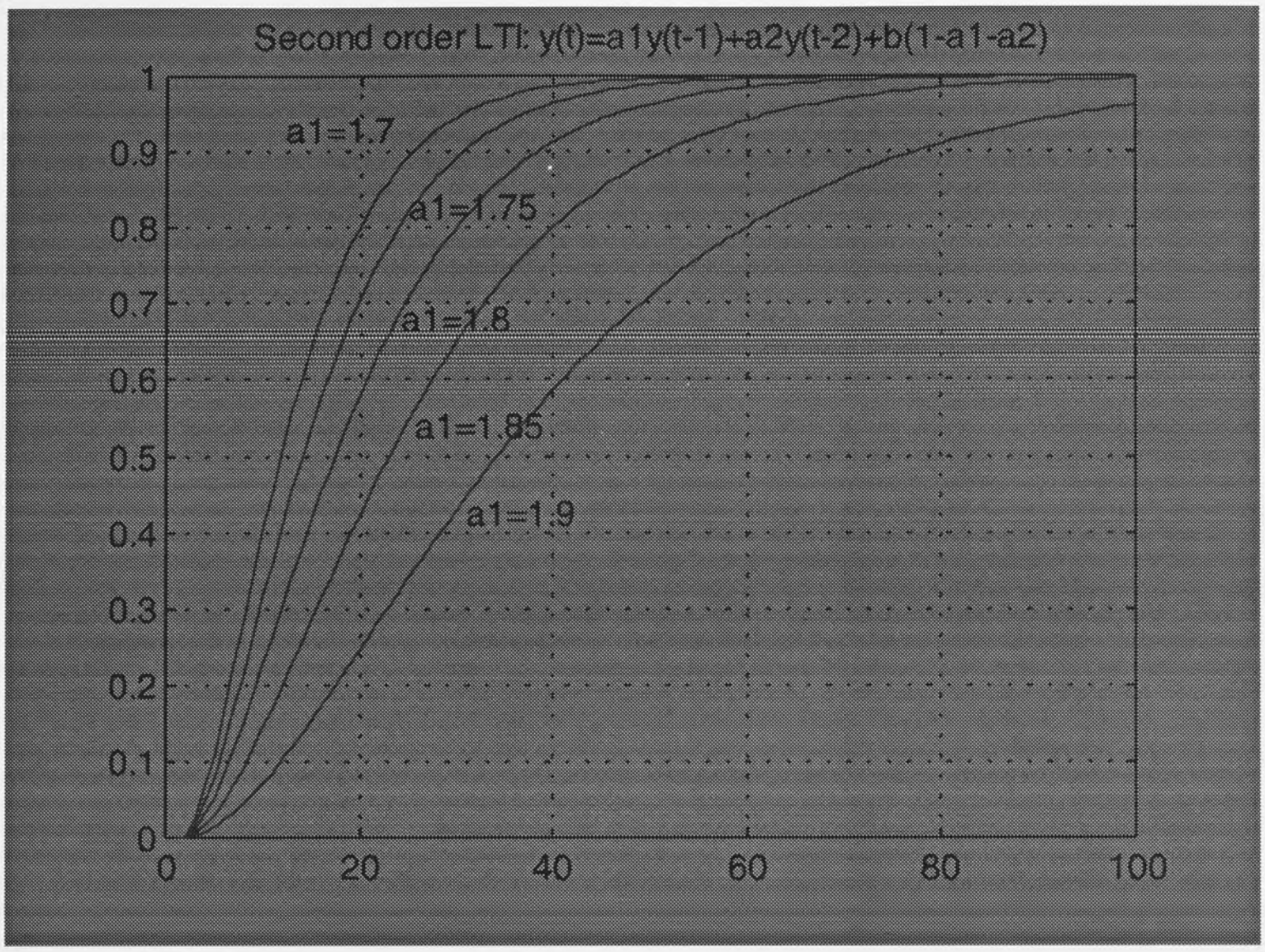


Figure 3C.3: A suite of curves of an LTI second-order system where $0 < a_1 < 2$; $b = 1$; $a_1^2 = -4a_2$; $y(1) = 0$; and $t = 1, 2, \ldots$.

CHAPTER 4

Dynamical Models for Plantation Forests

4.1 Introduction

Stand-based models that predict the effect of silvicultural activities on forest stands are vital in any forest management decision-making process. Due to the inherent complexity of management of intensively managed plantations *inter alia* the enormous variety of possible silvicultural regimes, the forest manager requires computer assistance to decide on the possible regimes that may be adopted and will meet the management objectives. Computer assistance, in the form of tools (collectively termed decision support systems), is described in chapters four and five. The tools are essentially based on a dynamic programming concept, where the optimisation is constrained by the dynamical models describing the forest growth response.

In the classical approach, model structures of the regression type that have been developed by other authors for different species are used and the coefficients estimated by means of the available data. In the dynamical systems approach, models have been identified *ab initio* and this chapter describes in detail their identification and validation.

4.2 Objective

A decision support system to assist forest managers should have the following features:

(a) an ability to *simulate* and *predict* growth and yield over a given time horizon for a given site;

(b) an ability to predict the *response* to thinning, potential losses to mortality and distribution of trees by species, size and quality;

(c) an ability to *partition* growth and yield into diameter classes and consequently estimate volume by class;

(d) an ability to *determine* the best thinning regimes on a specific site given management objectives, available markets and other economical forces; and

(e) easy to use , speedy in execution time and use average or total stand variables (i.e. data commonly available from plantation databases).

4.3 Dataset

The data used in this study were obtained from the *Pinus patula* Schl. et Cham., Correlated Curve Trend (CCT) spacing trials in Nelshoogte, in the Southern Transvaal region (30 deg 48 min E, 25 deg 48 min S) of the Republic of South Africa.

The tree species is of Mexican origin with a restricted natural distribution in the temperate humid regions of south-central Mexico. In these regions, the altitude ranges from 2300 to 2700 metres above sea level and the mean annual rainfall from 1000 to 1350 mm. Rain predominantly falls in the summer months. *P. patula* was introduced into South Africa in 1907 by Sir David Hutchins. In South Africa, the species grows best on cool, moist sites in the summer rainfall region and is planted on the humid seaward slopes of the escarpment and foothills from the eastern Cape Province to the northern Transvaal. It also grows well on the slopes of the higher coastal ranges and more humid parts of the interior plateau. The greater part of the area where it grows in South Africa lies between 900 and 1800 metres above sea level.

P. patula is the most important softwood plantation tree species grown in South Africa. Softwood species occupy 31% of the total area under commercial plantations, of which 246 000 ha (42%) is *P. patula* (Anon, 1988). Detailed descriptions of *P. patula*'s silvicultural and site requirements, can be found in Loock (1950), Poynton (1979), Schutz and Schafer (1985), Schonau and Grey (1987) and Perry (1991). Growth in the demand for softwoods is projected at 3.5% per annum. It is estimated that softwood will increase its share of the total roundwood market from the current 42% to 54% by the year 2010. The forecasted demand for softwood by 2010 will be approximately 14.8 million cubic metres (van der Zel, 1989). This increase in demand will place tremendous pressure on land acquisition for afforestation. However, due to the shortage of suitable land as well as increasing environmental pressures on potential afforestable areas, these requirements cannot be met solely through increased afforestation. The shortfall in demand will have to be met by maximising yield on existing sites. This can be achieved by implementing sound silvicultural practices such as optimum thinning strategies, the utilisation of genetically improved seed, fertiliser application and reduction in weed competition.

The CCT experiment was established in 1937 with four replications of each of the 16 treatments. The basic concepts of the CCT spacing experiments in South Africa were laid down by O'Connor (1935) and have been described in detail by various authors, such as Bredenkamp (1984). The trials were divided into two main experiments. One was to monitor the growth of unthinned stands, at a wide range of stand densities, and this was referred to as the basic series. In the second experiment, thinning strategies (intensity, timing and rate of thinning) were to be assessed.

Eight nominal stand densities (plots 1-8), ranging from 124 to 2965 stems per hectare, were established in the basic series. In order to avoid suppression by grass and weeds, the plots were planted at very high stocking levels (2965 stems/ha) and then thinned 'in advance of competition'. The schedule of stem number reduction

is given in Table 4.1. The thinning plots represent various degrees of growth suppression and release and can be classed into three groups:

(a) Plots 9-12. These four treatments are identical in terms of initial density, number of thinnings, degree of thinning and final stocking level. The only real difference is the age at which thinning commenced.

(b) Plots 13-14. These treatments represent similar degrees of suppression followed by different degrees of release.

(c) Plots 15-16. These treatments represent different degrees of suppression and release to similar densities.

Table 4.1

The schedule of the CCT trial in Nelshoogte expressed in residual stocking after a thinning treatment at a specific age.

Plot	Age (years)											
	0	1.67	3.50	4.00	5.00	6.00	7.00	8.00	10.67	15.17	19.25	23.33
1	2965											
2	2965	1483										
3	2965	1483	988									
4	2965	1483	988	741								
5	2965	1483	988	741	494							
6	2965	1483	988	741	494	371						
7	2965	1483	988	741	494	371	247					
8	2965	1483	988	741	494	371	247	124				
9	2965								1976	988	494	247
10	2965					1976			988	494	247	
11	2965	1976				988			494	247		
12	2965	988				494			247			
13	2965								988			
14	2965								494			
15	2965								494			
16	2965	988							494			

4.4 Method

The principal use of growth models is to simulate, predict and control growth and yield which are required for various aspects of forest management. At national or regional level, decision-makers formulating normative policies have to foresee the consequences of executing such policies. The consequences may be calculated by aggregating simulations and predictions of all the concerned stands (estate management) or simply confining the analysis to one stand with a set of proposed management rules.

With the aid of geographic information systems, it is now possible to visually interact with and interpret the forest growth and yield modelling results. Tools such as FORPLAN (FORPLAN Version 2: Users Guide, 1991) are used to generate linear programming matrices, using information derived from forest growth and yield projections, other forest values projections and geographic spatial data. Solutions to these linear programming formulations (FORPLAN matrices) are obtained by using techniques such as the simplex method. The linear programming analysis allows for a limited search of optimality, where long term goals are checked for their feasibility in fixed time intervals. Despite maximising timber production, other forest values such as biodiversity (Commonwealth of Australia, 1993), water quantity, visual impact and recreation, may have to be satisfied. Policies for estate management are influenced by the output from the optimisation procedure for the different *analysis areas* (FORPLAN Version 2: Users Guide, 1991; see figure 4.1). The Department of Natural Resources and Environment (in Victoria, Australia) has gone a long way in perfecting this tool for forest management (Lau, Vandenberg, and Willig, 1994).

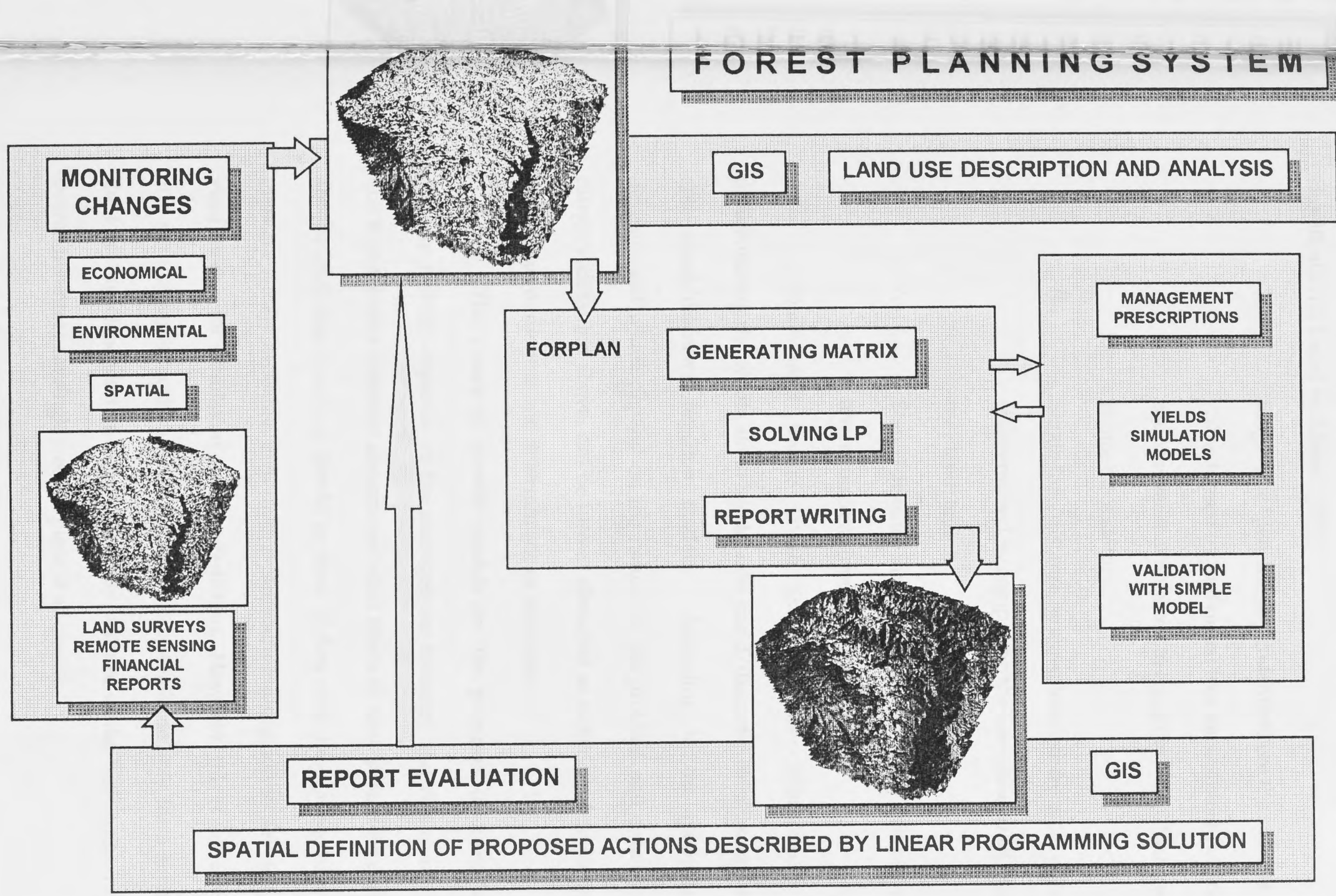


Figure 4.1: The functions of the forest planning, evaluation and monitoring system. It consists of five major components with the FORPLAN matrix generator as its engine. The GIS and remote sensing digital imagery provides the spatial framework for the LP model and also provides a means and the context for the visualization of the LP solution. (Lau, Vandenberg and Willig, 1994)

4.4.1 Growth model types

Growth models can be classified in a variety of ways. The following classification is based on Munro (1973):

(a) Single tree/distance dependent, where the stand is characterised by individual tree measurements, including tree coordinates that are used to compute individual tree competitive status;

(b) Single tree/distance independent, where the competitive status is determined by comparing the tree's size to all other trees in the stand; and

(c) Stand/distance independent, where only stand summary data are required as input.

Hann and Brodie (1980) extended this classification system by differentiating the whole stand model into stand/diameter class, stand/diameter-free and stand/diameter function models. According to the methods of model development which depend on the nature of the problem and data available, each category classified above, can be further classified as either mechanistic or black box, continuous or discrete and deterministic or stochastic.

The choice of growth models for the purposes of forest management decision-making depends on the management intensity and nature of the problem. The management intensity determines what kinds of data should be collected and if already available, how they should be used. If data with descriptions of single trees or plots are widely available, then there is sufficient information to develop growth and yield models for simulation and/or prediction. However, this is not always the case and at most times, such detailed information is available only for a small fraction of the forest in consideration. This is mainly because of the high costs involved in an overall inventory on single trees or plots of all stands.

Since most forest enterprises use stands or more accurately compartments as basic recording and management units and collect stand summary information for planning, it is reasonable that the whole-stand modelling approach would be ideal for forest management decision-making.

4.4.2 Model development

In model development much can be gained by defining precisely the managerial problem to be solved and limiting the model strictly to that problem.

Three elements in model development are observed: problem definition, specifying objectives, model building and model validation. The kind of model built is dependent on the problem definition and it also influences the manner in which the model will be implemented. Too little time is usually spent on precise problem definition. A well-defined forest management problem is more than half the modelling problem solved. For this study the objective is to develop a control model for determining optimal thinning regimes in the short-term (0-35 years) to satisfy management objectives.

Model building should be able to satisfy the basic criterion of *simplicity*. In general, simple models (or relationships) share one or more of the following features:

(a) easier to understand (i.e. transparent);

(b) characterised by fewer parameters (i.e. parametrically efficient);

(c) easier to test;

(d) require fewer inputs;

(e) easier to operate; and

(f) easily adjusted to fit new set of data (see chapter six).

Some of these features may be mutually exclusive and their presence or absence depends on the purpose of the model and how it will be used.

Despite the attraction of simplicity and its advantages, there is in science an underlying tension between simplicity and truth. Science is not interested in simplicity itself but only in so far as simplicity may constitute a means for forming and checking opinions.

Few realistic models have been employed as management tools in the natural sciences and this has been attributed to their complexity (Barlow, 1983; Cuff and Baskerville, 1983; Welch et al., 1981). Complexity impedes progress and it is the simpler models which are finding more acceptance as tools for management (Sands, 1988). It may be the case that simplicity is more acceptable in applied sciences than in research. Meisel and Collins (1973) noted that 'the growing accuracy and sophistication of models in many fields often leads, paradoxically, to limitations of their use'. Accordingly, there is considerable interest in procedures for simplifying complex models (Innis and Rexstad, 1983).

However, of equal importance is the appropriateness of the model. The context in which a model will be used, and who will use it, are major determinants of the nature of a model and of the modelling process. They also influence the simplicity of the resultant model. The important thing to note is that complexity and context are generally linked, i.e. context is the basis for deciding on the structure of the model; e.g. spatial or temporal resolution, the nature of the relationships in the models and the level of detail modelled.

Note that in the context of control i.e. maximising yield, the control model in this thesis may be much simpler than would be required if one is needed to simulate a plantation over all possible scenarios of operation. For example, it is unlikely that one would need to contend with explaining the impact of bushfires, soil erosion, climatic changes etc. Moreover the managed plantation has some level of error in modelling that may be tolerated as long as the silvicultural decisions are not affected. The control model is acceptable if the yield can be optimised correctly, and not

necessarily able to predict the forest plantation behaviour over extended periods of time (i.e. any predictions beyond 35 years).

Decision-making process

The decision-making process in any renewable resource management program, mainly relies on the biometric functions used which simulate growth and yield, and the ability to generate optimum or near-optimum management strategies. The stand management program in this chapter was developed with functions that take the form of dynamical models and a control formulation that enabled the choice of the 'best' silvicultural regime.

The resource management procedure that is generally followed by the decision-maker (at stand or estate level) is illustrated by the flow-chart in Figure 4.2. At the start of the procedure goals are set that are influenced by changes in site productivity and climatic conditions, the prevalence of pests and diseases, wood quality requirements, availability of markets, expected financial returns and government policy.

In the classical approach, expert knowledge is used to device a set of schedules (or management regimes) that may meet the desired goals, before or after an inventory of the forest resource. Only one schedule can be implemented and thus a further selection is carried out but this time using a decision support system (DSS). The DSS is used to simulate the growth of a stand or estate. A financial analysis can be done if the goal is to optimise returns. The schedule that maximises the benefits is chosen as the best. Further refinement is carried out due to other constraints, that could be budgetary, for instance. Finally, a practical schedule is arrived at that can be applied to the forest resource. At estate level the management procedure is complex because several stands of different age classes and productivity are dealt with.

The method in this chapter is used to illustrate how, at a silvicultural level an optimal stand regime can be chosen by directly translating the goals into an

optimisation formulation that can be solved by dynamic programming or maximum principle. Thus the flow-chart in Figure 4.2 would show a structure that starts from the goal setting to the DSS (that has an optimisation formulation) before or after a forest inventory, leading to an optimal regime. This approach may be used at an estate level if the models are available for the other sites (of different productivity) of the plantation estate.

4.5 Diameter Growth Modelling

Diameter growth within a stand is characterised in changes in the diameter distribution and is of major importance to the forest manager, for consequently determining the forest products by size and volume. Numerous models have been developed to characterise diameter distributions or to simulate diameter growth within stands. These diameter growth functions can be classified into two categories, Markovian and non-Markovian.

The Markov approach regards stand diameter growth as an aggregation of changes in diameter of many individual trees from one measurement to the next. In particular, it deals with the probability that a tree with an x_1 cm diameter at the beginning of a time interval will grow to x_2 cm at the end of that time interval. A Markov process is defined as one in which the probability of transition depends only on the initial state of the system at the beginning of a simulation period, and not on any previous state.

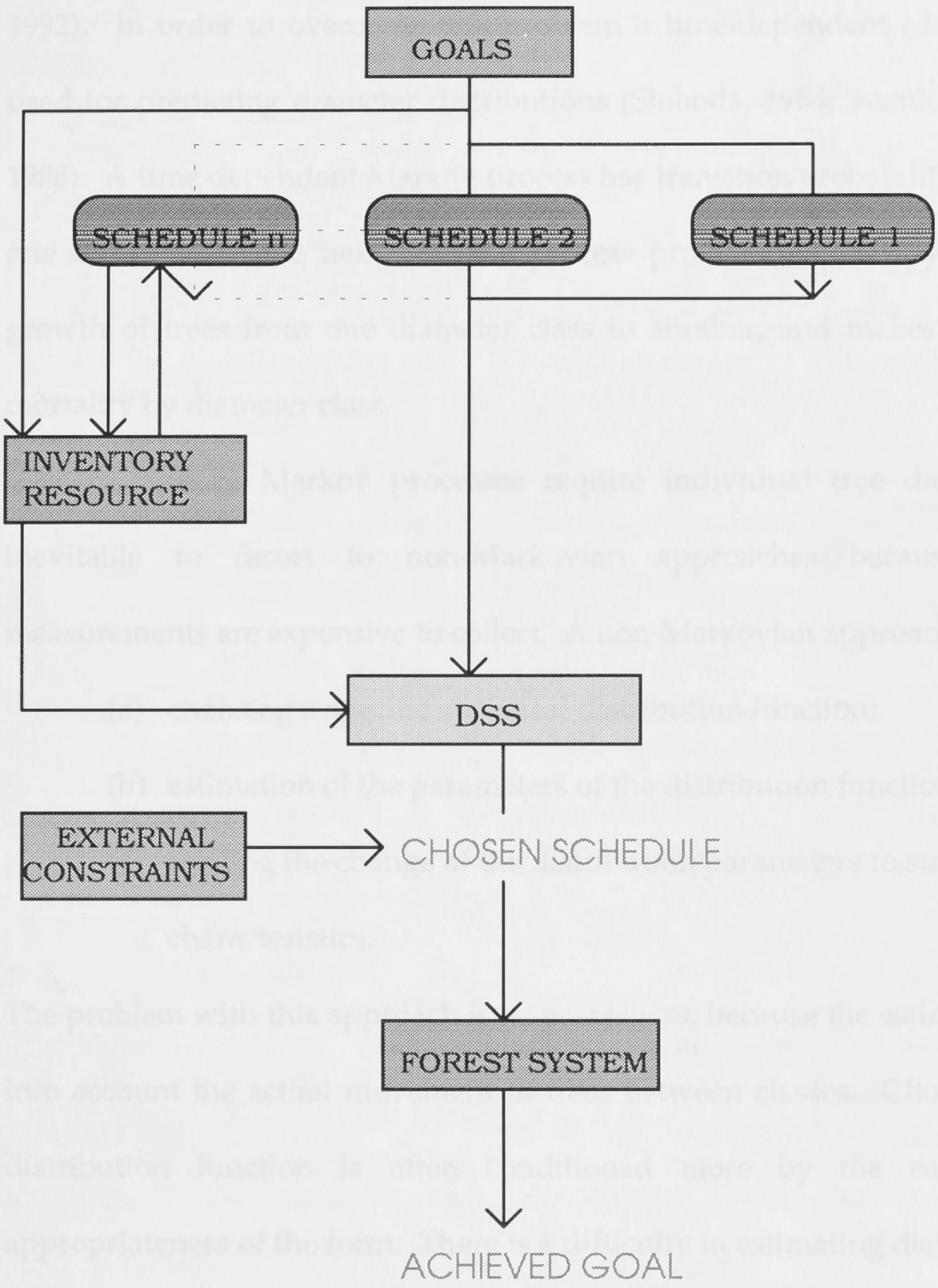


Figure 4.2: The flow of events in Resource management decision-making (Chikumbo, 1991).

Stationary Markov processes have been used to model diameter distributions, but found inadequate because prediction beyond one period results in a loss of accuracy, suggesting that the process may not have been stationary (Rong-wei, 1992). In order to overcome this problem a time-dependent Markov model can be used for predicting diameter distributions (Sloboda, 1984; Suzuki, 1971; and Tanaka, 1988). A time dependent Markov process has transition probabilities that change from one transition to the next. Such a process provides explicit dynamics of diameter growth of trees from one diameter class to another, and makes it possible to track mortality by diameter class.

Since Markov processes require individual tree data, it is sometimes inevitable to resort to non-Markovian approaches because individual tree measurements are expensive to collect. A non-Markovian approach involves:

(a) choosing a specific statistical distribution function;

(b) estimation of the parameters of the distribution function; and

(c) relating the change of the distribution parameters to stand characteristics.

The problem with this approach is its coarseness, because the estimation does not take into account the actual movement of trees between classes. Choice of the particular distribution function is often conditioned more by the ease of fitting than appropriateness of the form. There is a difficulty in estimating distribution parameters with any degree of accuracy (closeness to the truth) and precision (reproducibility of a result), (Chikumbo, 1991).

A regularly used distribution function in the non-Markovian approach is the Weibull function. It is widely used in forestry because it is very flexible and easily integrated. Treatments such as fertilisation may induce a shift in the population mean diameter and thinning will tend to skew the diameter distribution. The Weibull function is flexible enough to handle such variation (Little, 1976). However, there is no

particular reason for diameter distributions of trees to follow a Weibull or any other statistical function.

Diameter distribution models based on the Weibull distribution function require efficient procedures to determine the parameter estimates of its probability density function. Two basic approaches to determine the parameter estimates of the Weibull function are, the parameter prediction method (PPM) and the parameter recovery method (PRM). Both procedures are found lacking in one respect or the other. PPM has the advantage of using maximum likelihood estimation (MLE) which has many desirable statistical features and yields relatively accurate estimates of the Weibull parameters. PPM is well suited for unthinned stands. PRM has the advantage of simulating thinned stands but employs the method of moments which offers speed and simplicity in exchange for some loss in precision.

A third approach was devised by Chikumbo et al., (1992) which has the advantageous attributes common in PPM and PRM, namely the use of MLE for parameter estimation and the ability to simulate thinned stands respectively. The procedure is called the Dynamical Parameter Prediction Method (DPPM), because after estimating the Weibull parameters at the start of the projection period, say year one, parameters in the second and subsequent years are updated or estimated from linear dynamical models.

A series of equations are derived from integrating the Weibull into a dynamical diameter growth function, that will update the Weibull parameters at each time interval. The parameter a is initially estimated as half the minimum diameter in the stand and b and c by using MLE. Re-estimation of the parameters is only carried out after a thinning.

Having determined a control sequence or thinning strategy it is possible to determine the diameter class distributions and subsequently the volumes of the classes, by using the Weibull form to model the size class distributions. The stand

diameter growth dynamical model, provided it is first order, can be successfully linked to the Weibull function and used for distribution of growth by size classes (Chikumbo et al., 1992).

4.6 Model Structure Selection and Model Validation

Data from the Nelshoogte experiment were recorded at different sampling intervals and so mathematical manipulations were carried out to ensure equally spaced sampling intervals over the sampling period. The algorithm employed used linear interpolation and it required that the time-sampling vector from the source-data be monotonic. The cubic interpolation algorithm could have been used but required equally spaced sampling intervals from the source-data as well. The input variable for the modelling structure (2.11) was assumed constant between the sampling intervals and this assumption is quite common in computer controlled applications. Even if the input was not piecewise constant the assumption is still acceptable, provided the input does not change too much during a sampling period. The models identified were based on one year sampling intervals.

Figure 4.3 illustrates the computed one year sampling intervals for stand density (N) as 'o' and the superimposed measured stand density at unequal intervals '*', for a single replicate from each of the plots 1-5. The two sets of plots were quite comparable.

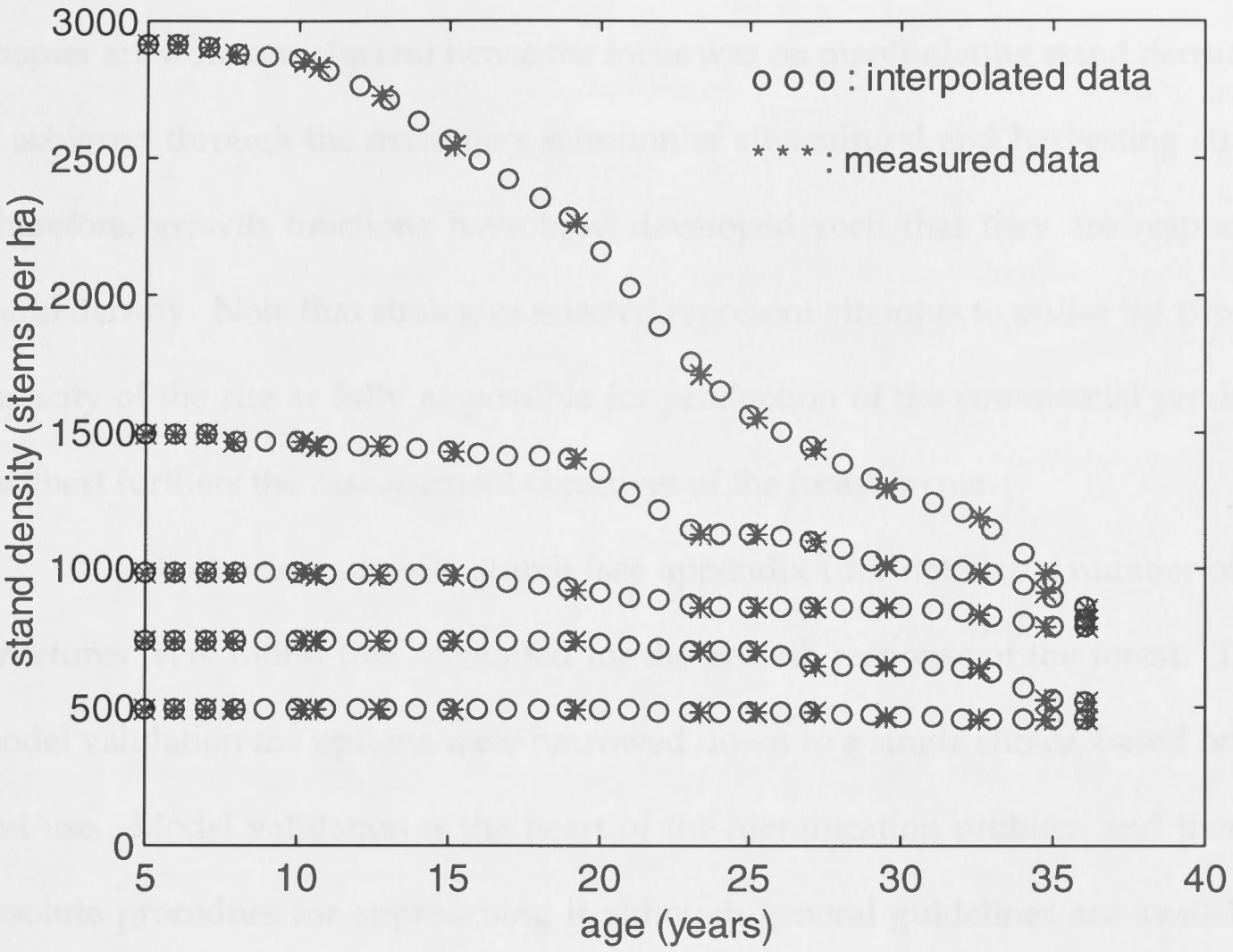


Figure 4.3 : Stand density interpolated data (at a sampling interval of one year) superimposed on the measured data (at unequal sampling intervals).

For some plots the time periods between measurements were considered too long for interpolation. Replicates with such gaps of missing data were not used for system identification but rather for *cross validation*. Cross validation checks the capabilities of a model to reproduce the observed trend when driven by input (independent) variables that were not used for the estimation of the model. This is a much tougher and more revealing test for model performance than statistical validation.

The growth rate achieved by any given forest is largely determined by two factors:

(a) the productive capacity of a site; and

(b) stand density.

These factors are subject to manipulation by the forest manager. The data used in this chapter are from one site and hence the focus was on manipulating stand density. This is achieved through the manager's selection of silvicultural and harvesting strategies. Therefore, growth functions have been developed such that they are responsive to stand density. Note that strategies selected represent attempts to utilise the productive capacity of the site as fully as possible for production of the commercial product mix that best furthers the management objectives of the forest owner.

Following a model search (see appendix I for details), a number of model structures were found that accounted for the growth response of the forest. Through model validation the options were narrowed down to a single choice, based on model end-use. Model validation is the heart of the identification problem and there is no absolute procedure for approaching it although general guidelines are available and have been given in chapter two.

4.6.1 Mortality function

When trees are planted and left without silvicultural treatment, competition for light, water and soil nutrients may cause some to die, i.e. stands experience mortality. In production forestry, it is essential to know the survival rates of any planting of trees. Such information is vital for forest planning where forecasts are made for timber volume. Thinning is designed to anticipate mortality and reduce competition. Figure 4.3 shows typical survival of unthinned stands of *P. patula* (for the Nelshoogte CCT trial) where stands with initial stand densities greater than 1500 stems/ha suffered significant mortality in the first 15years, a time period before which the first thinning is normally carried out. Most initial planting densities for timber production plantations would not exceed 1500 trees/ha (Shepherd, 1986). Therefore, modelling of mortality was not pursued.

4.6.2 Basal area function

The identification process for the nonlinear basal area function was a two-step approach (Chikumbo and Mareels, 1995b):

(1) a first order, linear time invariant model that explained the general basal area growth trend was identified for each plot;

(2) the parameters of the models in (1) were found to have a consistent relationship with the initial density. Therefore, these parameters were modelled using polynomials that were functions of stand density.

Eight basal area models were developed from the first eight plots (1-8) using a single replicate from each plot for system identification and the remainder of the replicates used for cross validation for each plot. The statistical validation of these models are given in Table 4.2. They show good statistical properties. In other words, 1/4 of the data were used for system identification and the remaining 3/4 used for cross validation. The estimated parameters from these eight models were used to fit polynomials that were functions of stand density in the range of 124-3000 stems/ha and hence a general form of the basal area function (i.e. model (4.1)) was developed. Figures 4.4-11 show a replicate from each plot (for plots 1-8) that was used for cross validation. Table 4.3 shows the mean squared errors of these cross validation plots which indicate the robustness of model (4.1) across the 124-3000 stems/ha stand density range. The correlation function of the residuals and the cross-correlation of the residuals and input variable to model (4.1) indicated an unbiased model that was representative of the observed basal area trend within 99% confidence limits (see appendix I for the MatLab correlogram output). The model was as follows:

$$BA(t) = aBA(t\text{-}1) + b \tag{4.1}$$

where

BA = stand basal area (m^2/ha)

$$a = 0.93 + 0.01\frac{x}{1000} - 0.047\left(\frac{x}{1000}\right)^2 + 0.01\left(\frac{x}{1000}\right)^3; \quad (4.2)$$

$$b = 2.32 + 4.24\frac{x}{1000} - 0.35\left(\frac{x}{10000}\right)^2; \quad (4.3)$$

x = initial density (stems/ha).

Table 4.2: Statistical validation of the BA models identified for plots 1-8.

PLOT REPLICATES	*INITIAL DENSITY (stems/ha)*	*MEAN SQUARED ERROR*	*CORRELATION COEFFICIENT*	*RESIDUALS*
1A	2916	0.10	0.99	horizontal band
2A	1483	0.15	0.99	horizontal band
3A	988	0.003	0.99	horizontal band
4A	741	0.56	0.998	horizontal band
5A	494	0.38	0.99	slightly increasing trend
6A	371	0.005	0.99	slightly increasing trend
7A	247	0.10	0.99	horizontal band
8A	124	0.88	0.99	horizontal band

Table 4.3: Mean squared errors from cross validation with BA model (4.1).

PLOT REPLICATES (not used in model development)	*INITIAL DENSITY (stems/ha)*	*MEAN SQUARED ERROR (for basal area prediction)*
1D	2792	0.07
2B	1483	0.0
3C	988	0.06
4C	741	0.25
5D	494	2.37
6C	371	0.08
7B	247	0.35
8D	124	0.96

All the predictions for cross validation start at the ages of five or eight years for either of the two reasons:

(1) other growth variable measurements, with the exception of stand density, were not available until age five; and

(2) All 'thinning before competition' treatments were completed at age eight.

The above situation applied to the data used for developing all the models in this chapter and cross validations for stand basal area, mean stand height and volume.

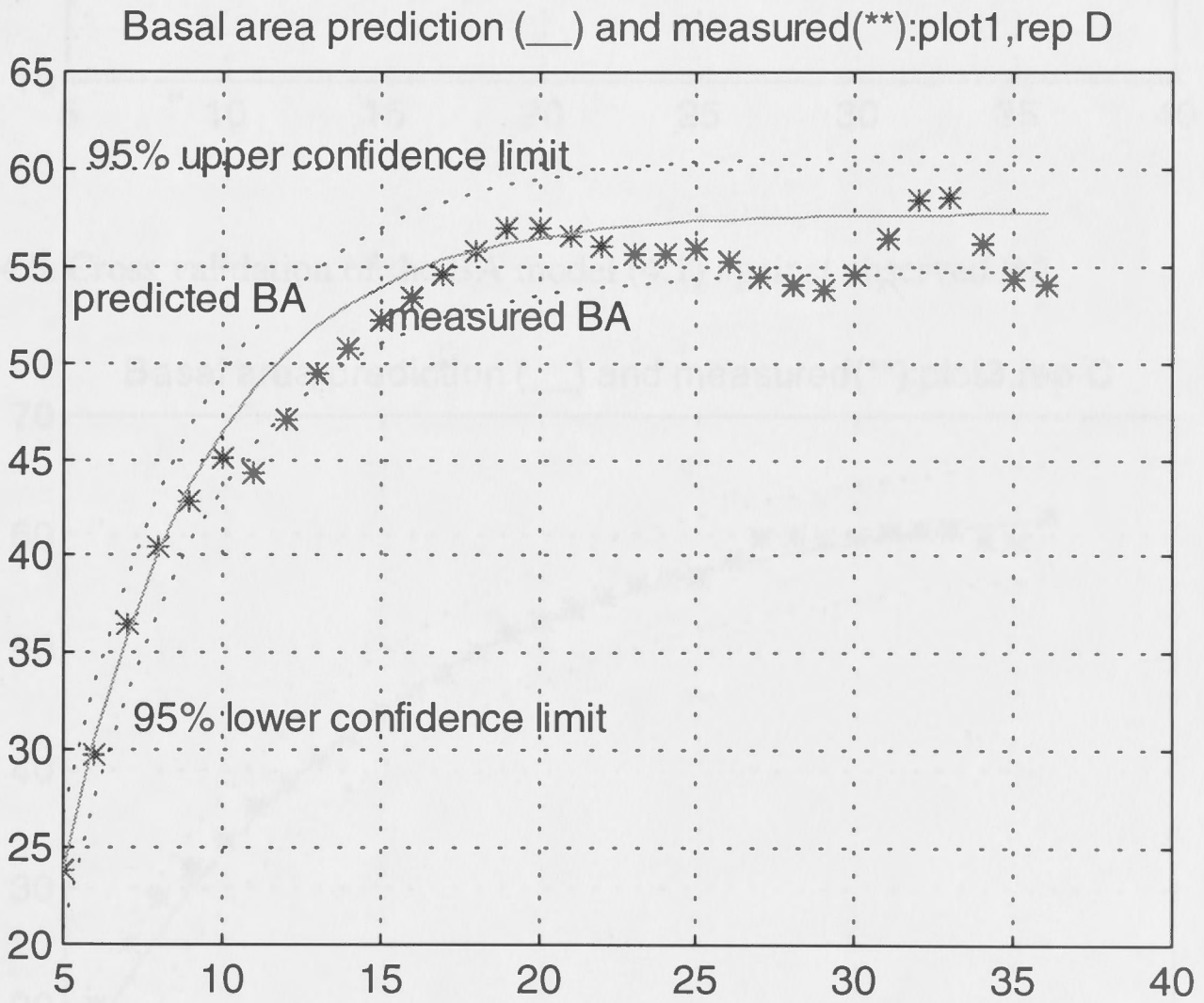


Figure 4.4: Cross validation of the BA model (4.1) against observed BA

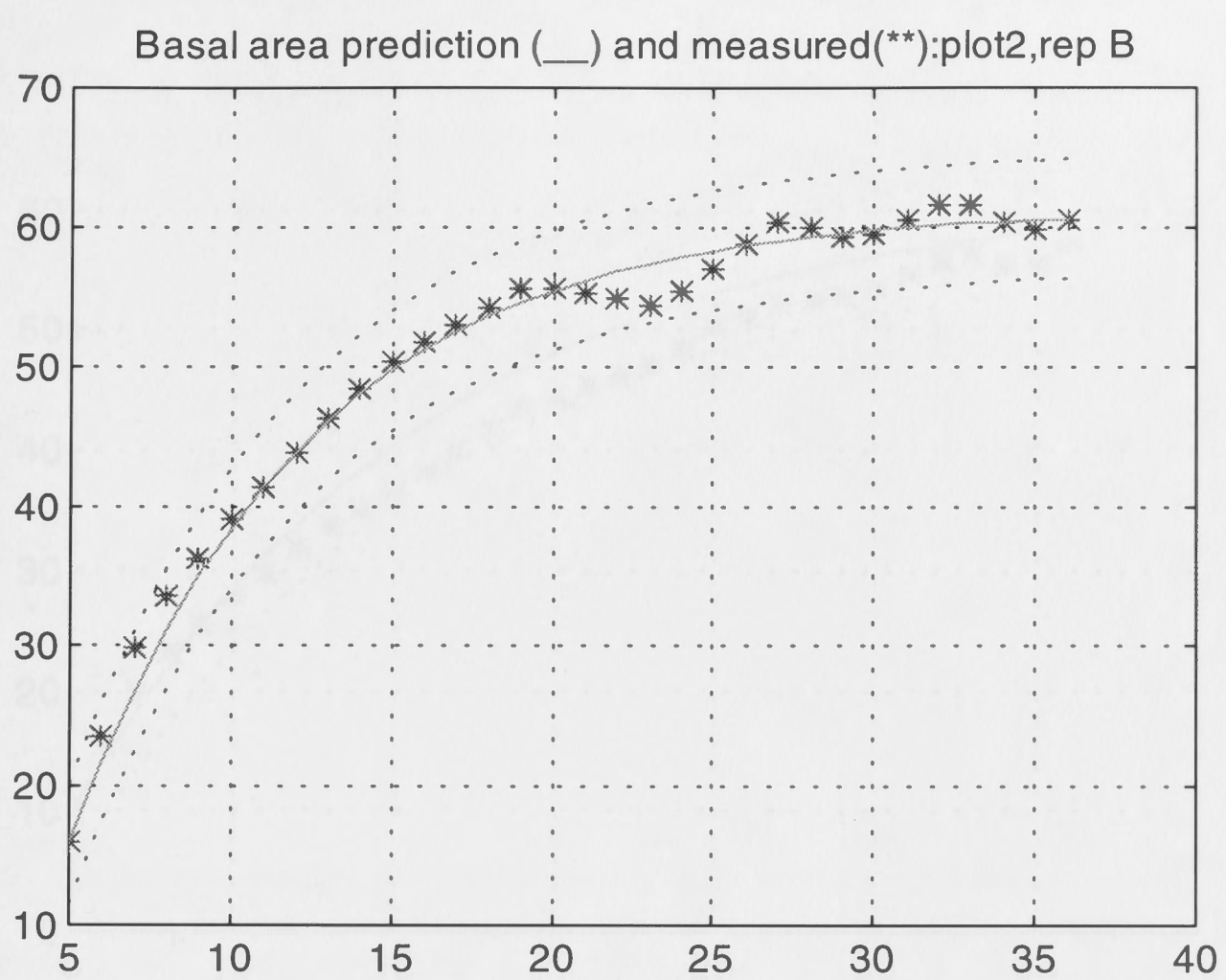


Figure 4.5: Cross validation of the BA model (4.1) against observed BA

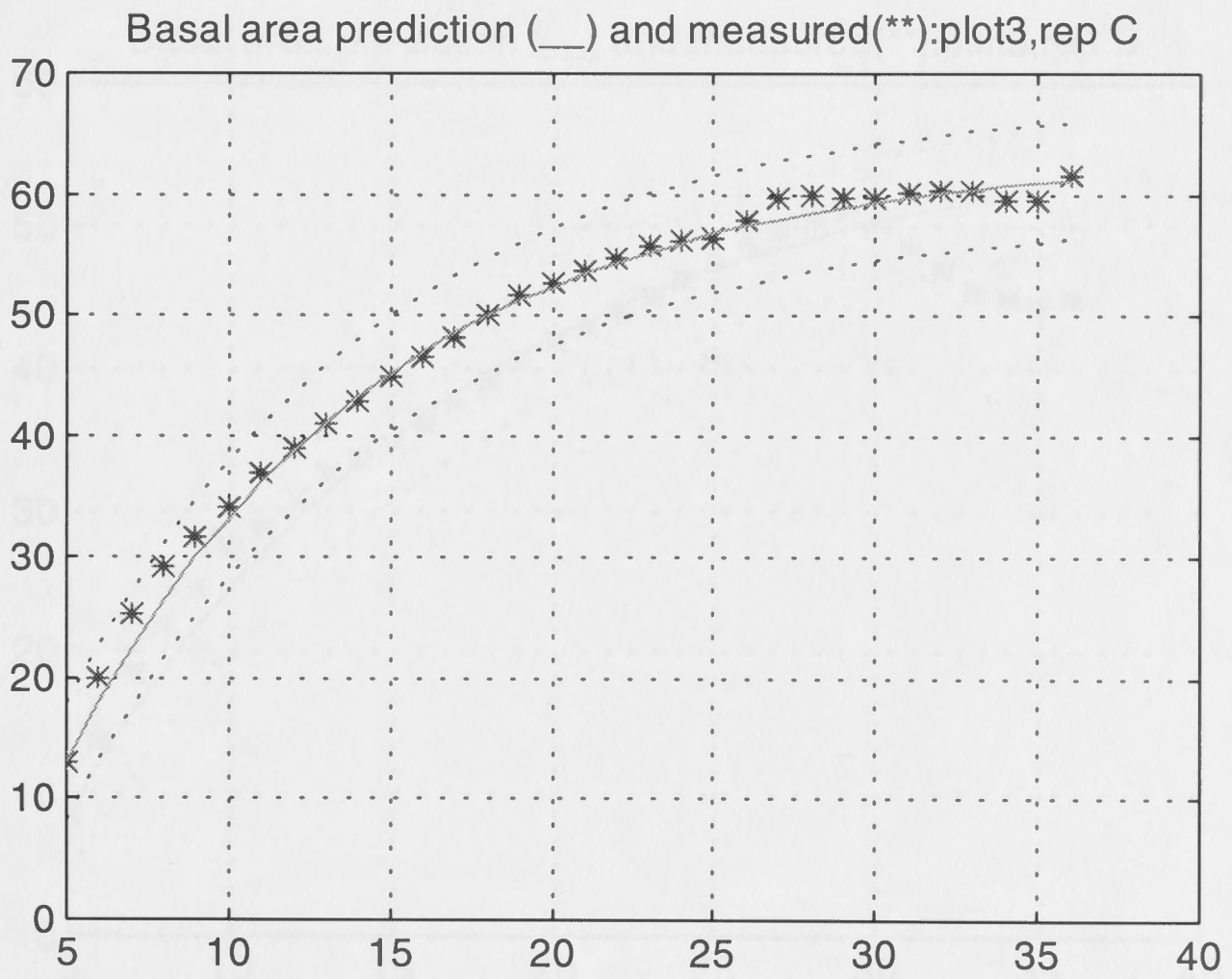


Figure 4.6: Cross validation of the BA model (4.1) against observed BA

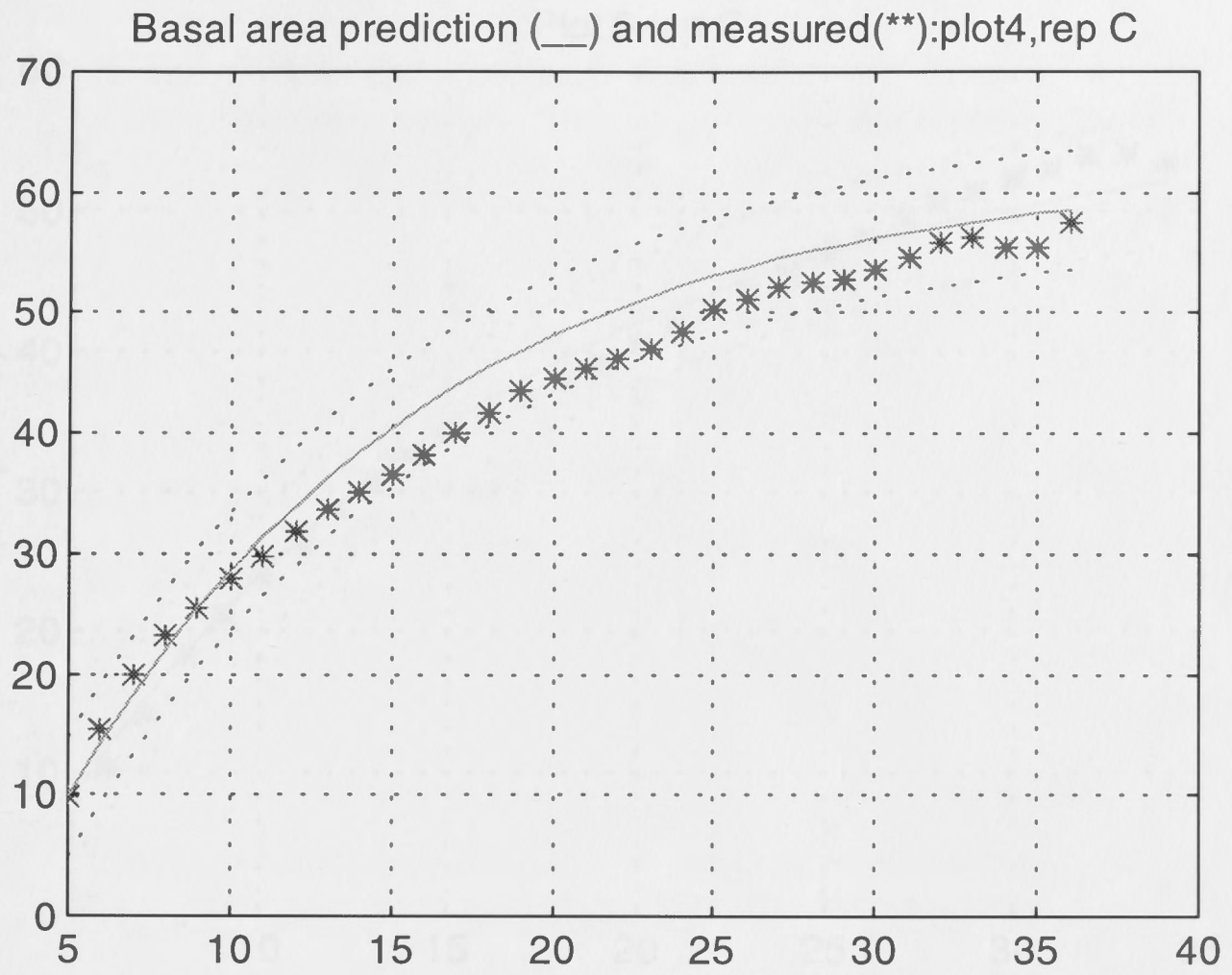


Figure 4.7: Cross validation of the BA model (4.1) against observed BA

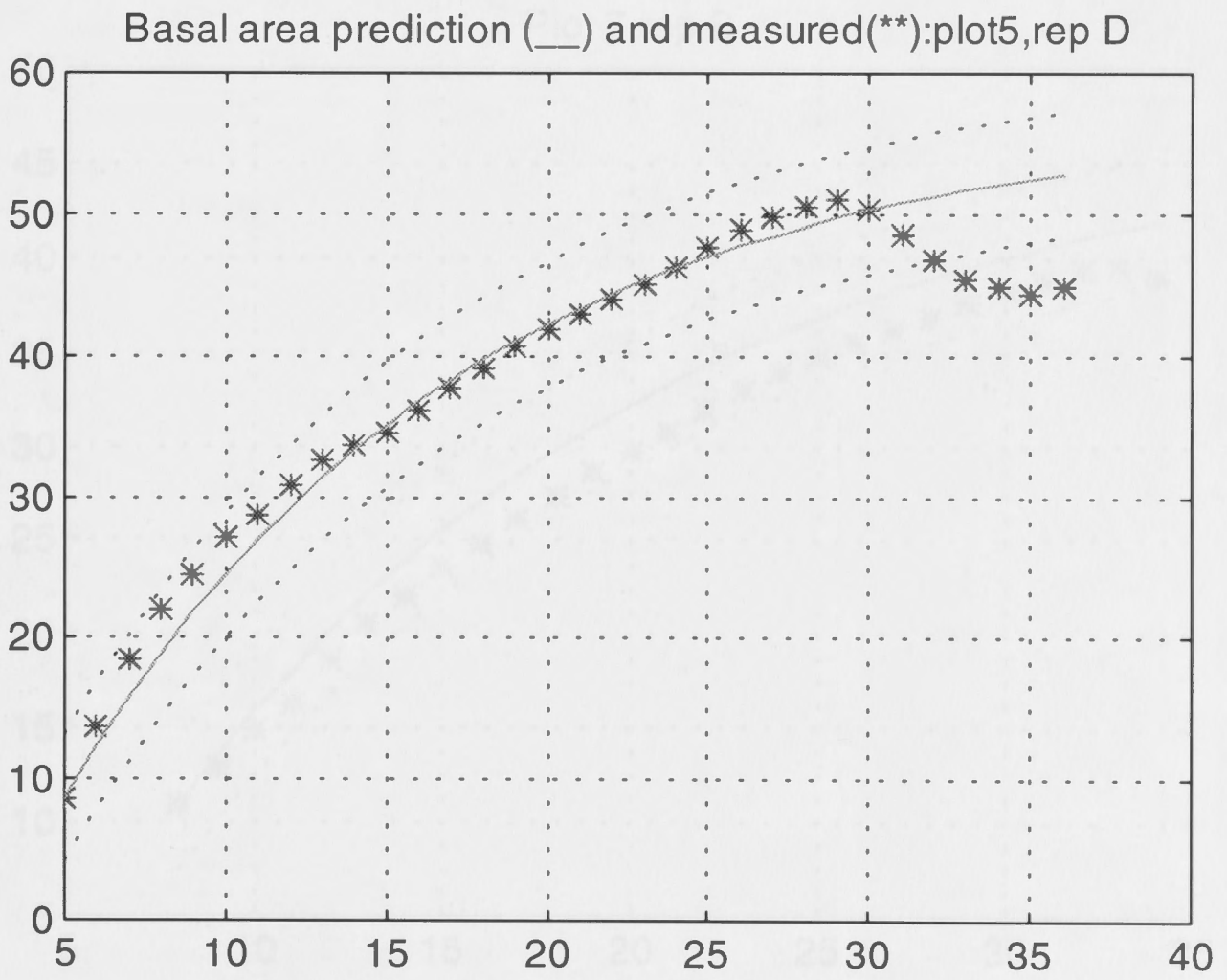


Figure 4.8: Cross validation of the BA model (4.1) against observed BA

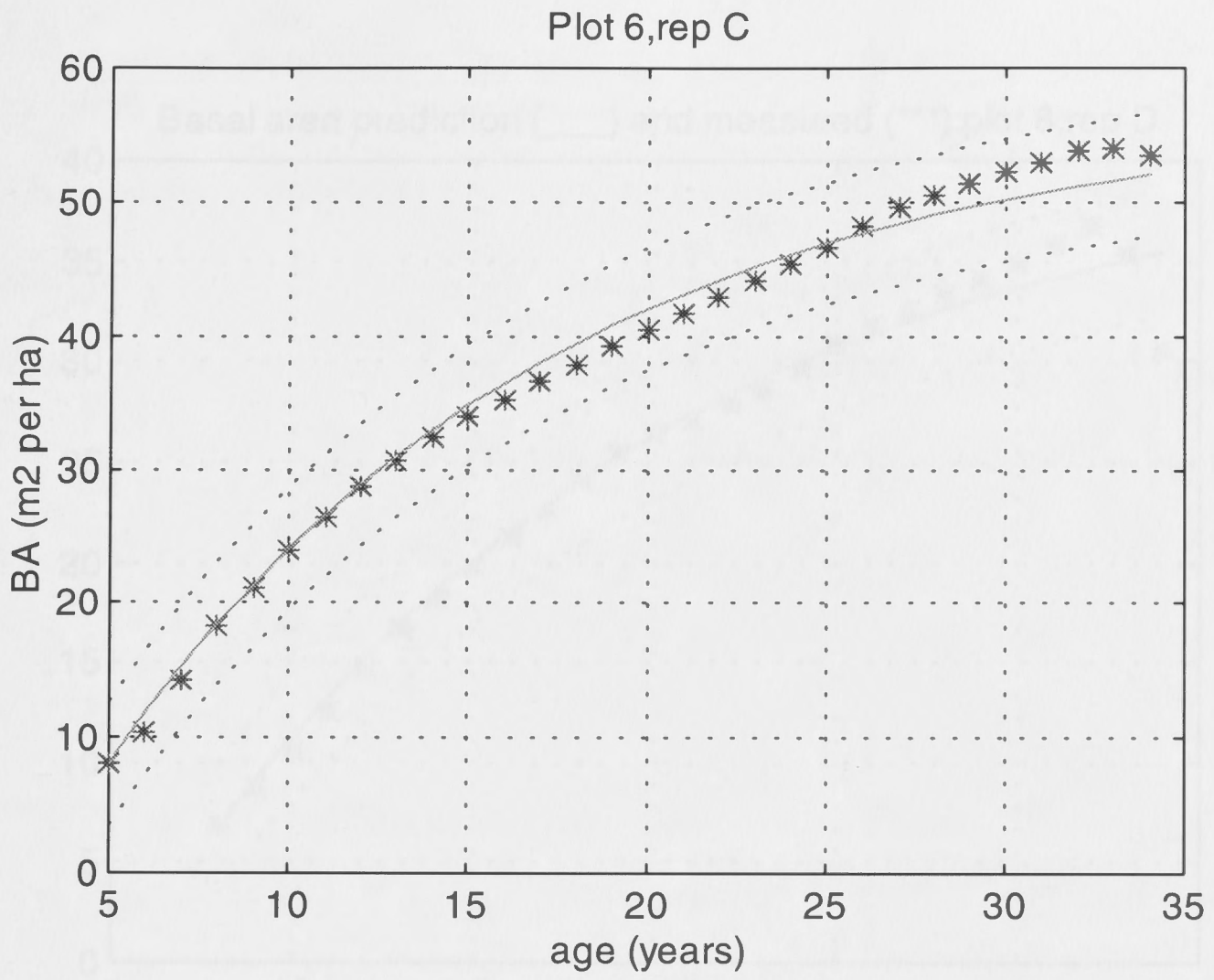


Figure 4.9: Cross validation of the BA model (4.1) against observed BA

Figure 4.10: Cross validation of the BA model (4.1) against observed BA

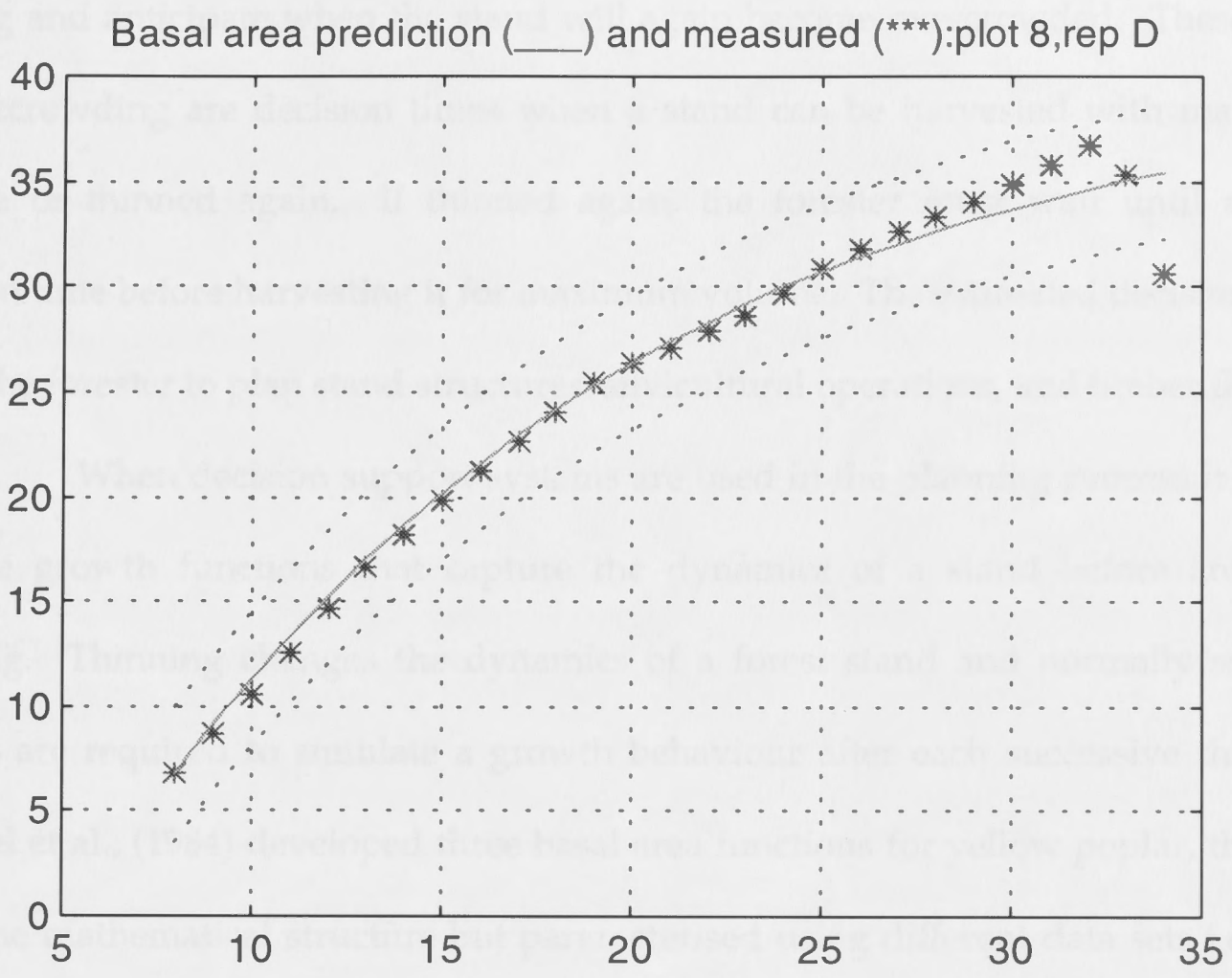


Figure 4.11: Cross validation of the BA model (4.1) against observed BA

4.6.3 Thinning responses

Thinning has always been one of the most important silvicultural alternatives because thinning can influence growth through controlling competition, potential volume losses to mortality, and distribution of trees by species, size and quality. Quantitative models are required for predicting growth of thinned stands and evaluating effects of various thinning strategies. Growing space occupied by a tree is suddenly made available to the surrounding trees which expand to refill it. How rapidly the surrounding trees refill the growing space depends on a number of factors such as the rate of crown expansion that depends on the innate crown and root characteristics, tree age, site characteristics, tree vigour and amount of growing space released (Oliver and Larson, 1990).

The forester can plan different thinning regimes based on the current tree spacing and anticipate when the stand will again become overcrowded. These times of overcrowding are decision times when a stand can be harvested with maximum volume or thinned again. If thinned again, the forester must wait until another decision time before harvesting it for maximum volume. The estimated decision times allow the forester to plan stand structures, silvicultural operations, and timber flows.

When decision support systems are used in the planning process it is vital to have growth functions that capture the dynamics of a stand before and after thinning. Thinning changes the dynamics of a forest stand and normally separate models are required to simulate a growth behaviour after each successive thinning. Knoebel et al., (1984) developed three basal area functions for yellow poplar, that had the same mathematical structure but parameterised using different data sets i.e., data for growth before thinning, after the first thinning and after the second and subsequent thinnings.

Knoebel et al., (1984) were able to demonstrate that the functions responded to:

(a) the intensity of thinning;

(b) number of thinnings in a single rotation; and

(c) timing of thinning,

by simulating two fictitious regimes in each case, that differed in basal area removed at each thinning; the number of thinnings in each rotation for different thinning strategies; and timing of the thinnings. However, no cross validation was done, that would have revealed the capability of the basal area functions to describe fresh datasets from a thinning process.

Pienaar (1979) asserted that if a plantation density of say 2000 stems/ha was thinned to 1000 at age 6 years and there was no difference in growth response between the remaining 1000 and those in an unthinned plantation on the same site

which was also 6 years old and had 1000 stems/ha, it would seem reasonable to expect the growth after thinning to equal growth in the unthinned plantation after age 6. The assertion could only hold if the thinning operation was carried out in a manner that justified the assumption that removal was not selective for size.

In a second scenario, Pienaar (1979) asserted that if a planting density of 2000 was thinned to 1000 at age 15 years and there was a difference in growth response between the remaining 1000 and those in an unthinned plantation of the same age, site index and density, the difference would reflect the relative degree to which the remaining trees would have been affected by suppression, relative to those in the unthinned plantation. He finally concluded that it would no longer be reasonable to equate growth after thinning to future growth in the unthinned counterpart. The problem was then to develop models that would reflect this assertion. The conventional regression-type functions generally used to model say basal area growth, usually have no ability to recognise a change in the dynamics of a plantation due to suppression and release (see chapter two).

By applying dynamical models only initial densities and initial basal areas (at the start of a projection) matter in the comparison with the unthinned stand basal area responses. This is because the models are time functions and dynamically predict the behaviour of a growth response based on the previous observations. Pienaar's conclusion in the second scenario has led to the development of different growth regression models between thinnings.

Chikumbo and Mareels (1995b) used the BA function (4.1) to predict BA responses following thinning at different intensities and timing. Prolonged periods of BA measurements, between any two successive thinnings from replicates in plots 9-16 were interpolated to one-yearly sampling intervals and used for cross validation.

Chikumbo and Mareels (1995b) also compared function (4.1)'s performance against a nonlinear continuous-time BA model and a multiple regression BA predictive model that were developed by Harrison et al., (1994):

$$BA = 57.1609[1 - e^{-\beta t}]^{1.95} \tag{4.4}$$

where

$\beta = 0.0044x^{0.4977}$

x = planting density (trees/ha)

and

$$ln(BA_2) = ln(BA_1) - 34.0846\left(\frac{1}{t_2} - \frac{1}{t_1}\right)$$
$$\ldots + 0.1717(ln\,x_2 - ln\,x_1) + 0.518(ln\,HD_2 - ln\,HD_1) \tag{4.5}$$
$$\ldots + 2.9567\left(\frac{ln\,x_2}{t_2} - \frac{ln\,x_1}{t_1}\right) + 4.3352\left(\frac{ln\,HD_2}{t_2} - \frac{ln\,HD_1}{t_1}\right)$$

where

ln = natural logarithm

x_1 = stand density at start of projection period

$$x_2 = x_0 - (x_0 - x_\infty)[1 - e^{-0.0001598t^{2.0175+0.0001588x_0}}]$$

x_0 = planting density (trees/ha)

x_∞ = asymptotic density (fixed at 90 trees/ha)

HD = average dominant height (defined as the average predicted height of trees in the upper quintile of the diameter distribution)

$$HD_2 = HD_1\left[\frac{1 - e^{-0.0482t_2}}{1 - e^{-0.0482t_1}}\right]^{0.9446}$$

Harrison et al., (1994) pooled all the replicates from plots 1-8 to develop models (4.4) and (4.5). The shape parameter β, of model (4.4) was found to have a consistent relationship with stand density and therefore was modelled as a function of stand density for the range 124-3000 stems/ha. The correlation coefficient and mean squared

error of the overall model (4.4) were *0.9567* and *3.86* respectively (Harrison et al., 1994). Harrison et al., (1994) did not make any comments on the distribution of the residuals for model (4.4) but mentioned that model (4.4) would be suited for evaluating different management regimes for unthinned plantations. The reason why model (4.4) was compared against models (4.1) and (4.5) was because it was developed from the same data set as models (4.1) and (4.5) and that its shape parameter was modelled to respond to changing initial stand density.

Model (4.5) is a *projection* model that was derived from a BA *prediction* model that had a correlation coefficient of *0.987* and a mean squared error (expressed as a percentage) of *12.0%*. Model (4.5) was developed to ideally suit future yield projections for any planting densities within the range of treatments in the Nelshoogte CCT trial. This entailed predicting the number of surviving trees and average dominant height. The survival equation based on the Weibull distribution function had a correlation coefficient and mean squared error of *0.9839* and *85.27* respectively. The average dominant height had a correlation coefficient of *0.9622* and a mean squared error of *1.32*. The residuals of the overall model were well distributed but variation increased in predicted BA over *50* m^2/ha.

The modelling approach used by Harrison et al., (1994) is different from the dynamical modelling one where model development involved two stages:

(1) model selection (that includes statistical validation) using eight replicates, one from each of the eight plots; and

(2) cross validation using the remainder of the replicates from plots 1-8.

This approach should not be confused with the two-stage model identification process that was used to determine the BA model (4.1) which falls in the first stage of this dynamical modelling approach. The dynamical modelling approach can be difficult to implement when data are scarce, but attempts should be made at all cost to have some

data set aside for cross validation because it is as much part of model development as the first stage. Because of these differences in the modelling approaches it becomes difficult to compare a dynamical model with a conventional model (even in situations where the same data have been employed for model development) by simply looking at the statistical validation; independent data have to be used to test the performances of the models by using a common statistical measure.

Note that the reason behind testing models (4.1), (4.4) and (4.5) for their performance against thinned stands is that the modelling data from plots 1-8 reflected the growth responses over a range of densities (124-3000 stems/ha) which was achieved by 'thinning before competition' (see Table 4.1). Therefore, it would be expected that the models would respond to changes in stand density over the range of 124-3000 stems/ha. The BA function (4.1) performed distinctly better and showed higher accuracy than models (4.4) and (4.5) (Chikumbo and Mareels, 1995b).

Mean squared error was used to compare the three models and a summary is given in Table 4.4. Model (4.1) showed, on the average, consistently lower mean squared errors than the other two models. Model (4.4) had the lowest mean squared error in one case and a mean squared error that exceeded the variance of the observed data in two cases. Model (4.5) was not reliable and in six cases out of nine had a mean squared error that exceeded the variance of the observed response. The plots of this validation (in Table 4.4) for these BA models are in Figures 4.12-20.

From Figures 4.12-20, model (4.4) seems to have the right shape but has an inability of determining the right scale of basal area response after thinning. In situations where the cross validation starts from a young age (3-11 years), model (4.4) will generally produce an accurate prediction confirming that the model is suited for predicting basal area in unthinned stands. Model (4.5) seems to predict the correct shape but tends to decay faster than the model (4.4) and also has a scale problem. When model (4.5) is initialised with an observed residual basal area after thinning, it

has no ability of responding at the expected scale (see Figures 4.12-20). However, if the residual basal area is predicted from the prediction version of model (4.5) (which tends to be higher than the observed basal area from the validation plots in Figures 4.12-20) then its response is closer to that of model (4.4).

In contrast, model (4.1) has the ability to change shape and scale depending on the initial density. As a result model (4.1) is responsive to thinning. This validation demonstrates that model (4.1) has a structure that can reflect the effects of thinning and is thus suited to predict basal response in thinned stands.

Table 4.4: Calculated mean squared errors from cross validation data of the models (4.1), (4.4) and (4.5). The lowest mean squared value for each test is shown in bold print and any that is greater than the variance of the observed data are shown in italics.

PLOT REPLICATES	*AGE RANGE*	*INITIAL DENSITY*	*MEAN SQUARED ERRORS*		
		(stems/ha)	*(of BA)*		
			model (4.1)	model (4.4)	model (4.5)
9D	24-36	247	**1.19**	17.81	*99.47*
11B	16-36	247	3.13	**0.4**	*102.98*
12C	11-34	247	**3.23**	3.35	65.30
13C	11-29	988	**18.53**	80.23	*291.56*
14B	13-29	482	**1.46**	22.53	*198.79*
15B	11-36	494	**0.95**	44.53	*422.53*
15D	11-34	494	**14.98**	*112.73*	*572.48*
16B	3-10	988	**3.08**	19.65	85.84
16B	11-36	494	**6.80**	13.59	*134.94*

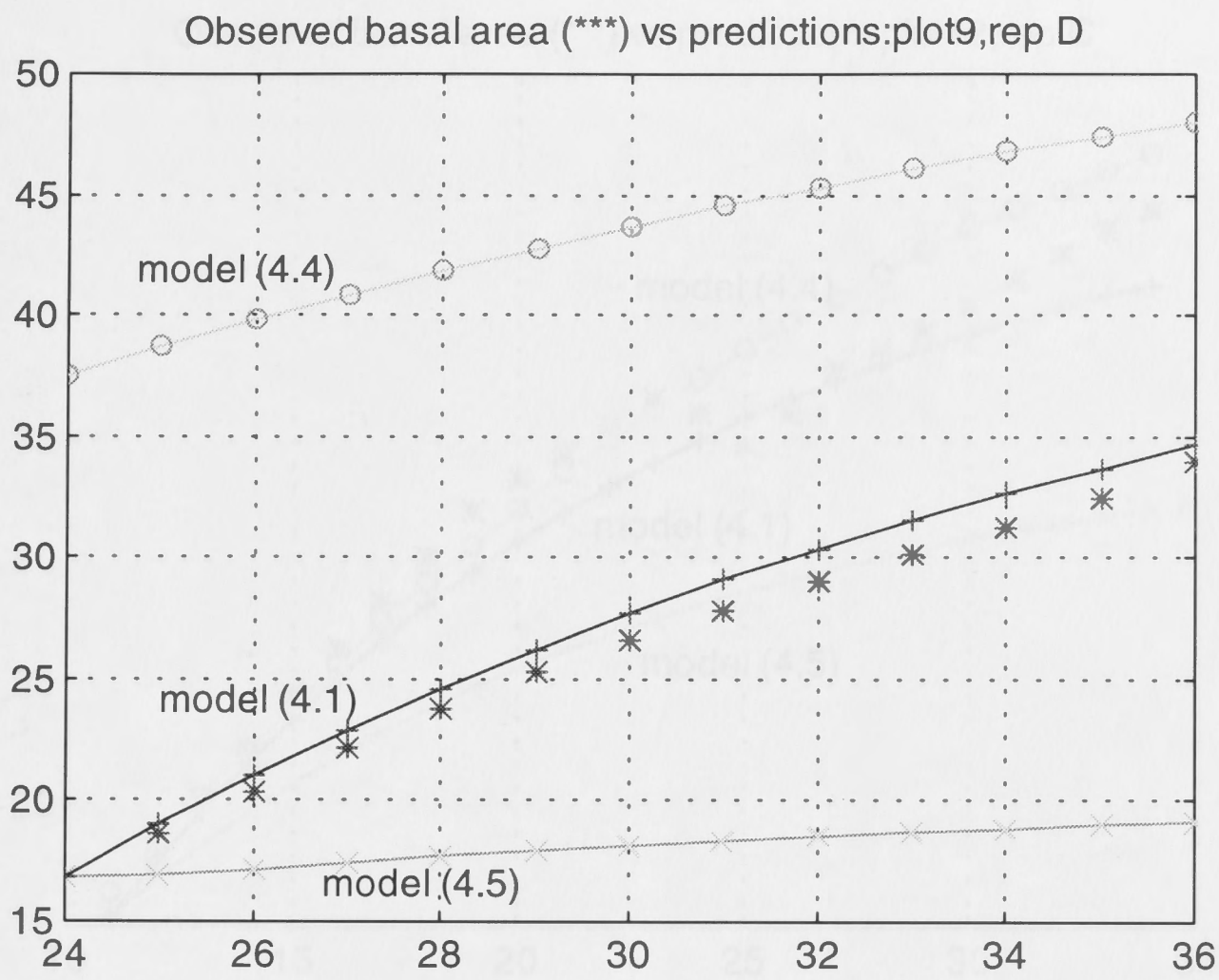


Figure 4.12: Thinning response at a residual density of 247 stems/ha at age 24.

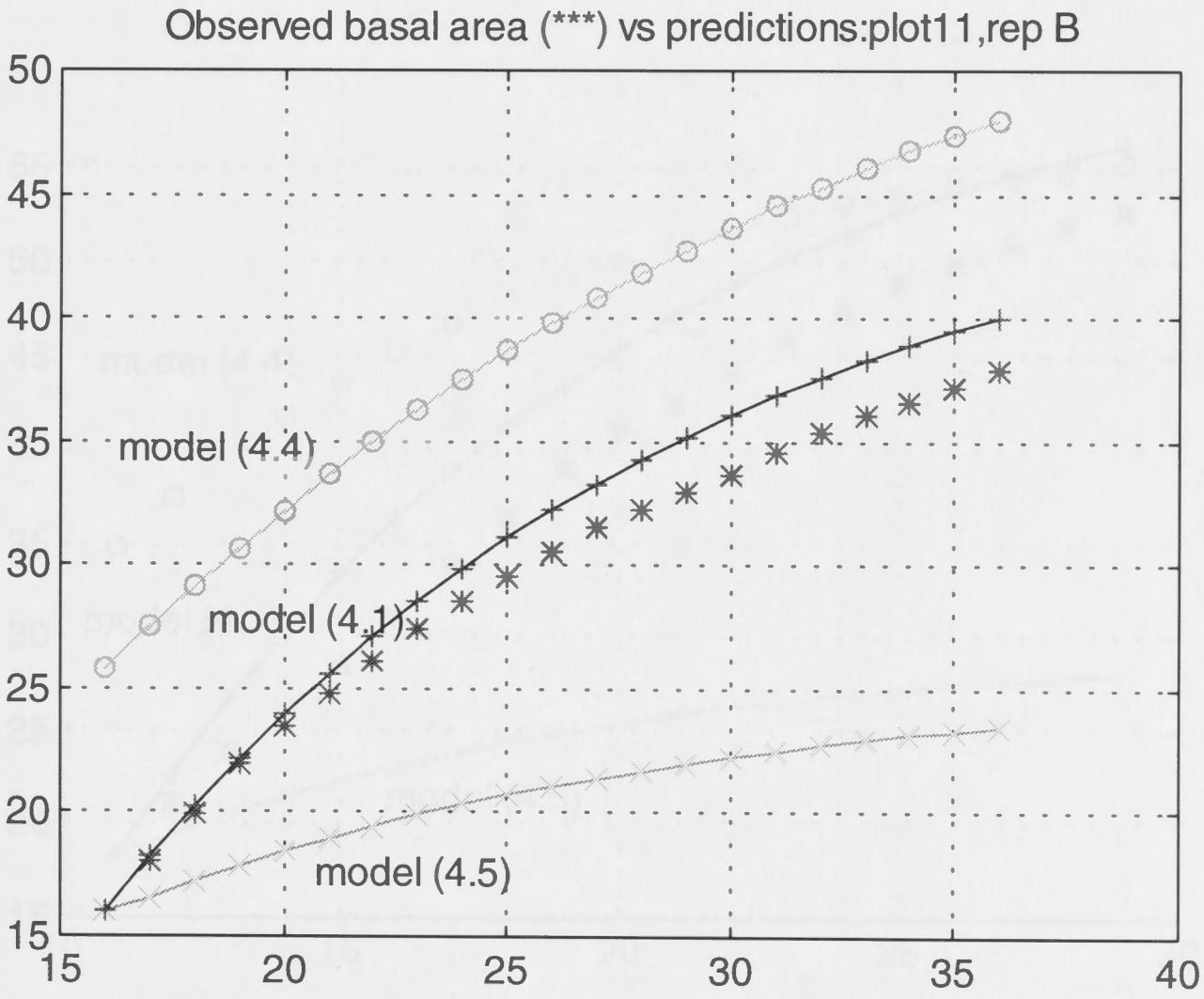


Figure 4.13: Thinning response at a residual density of 247 stems/ha at age 16.

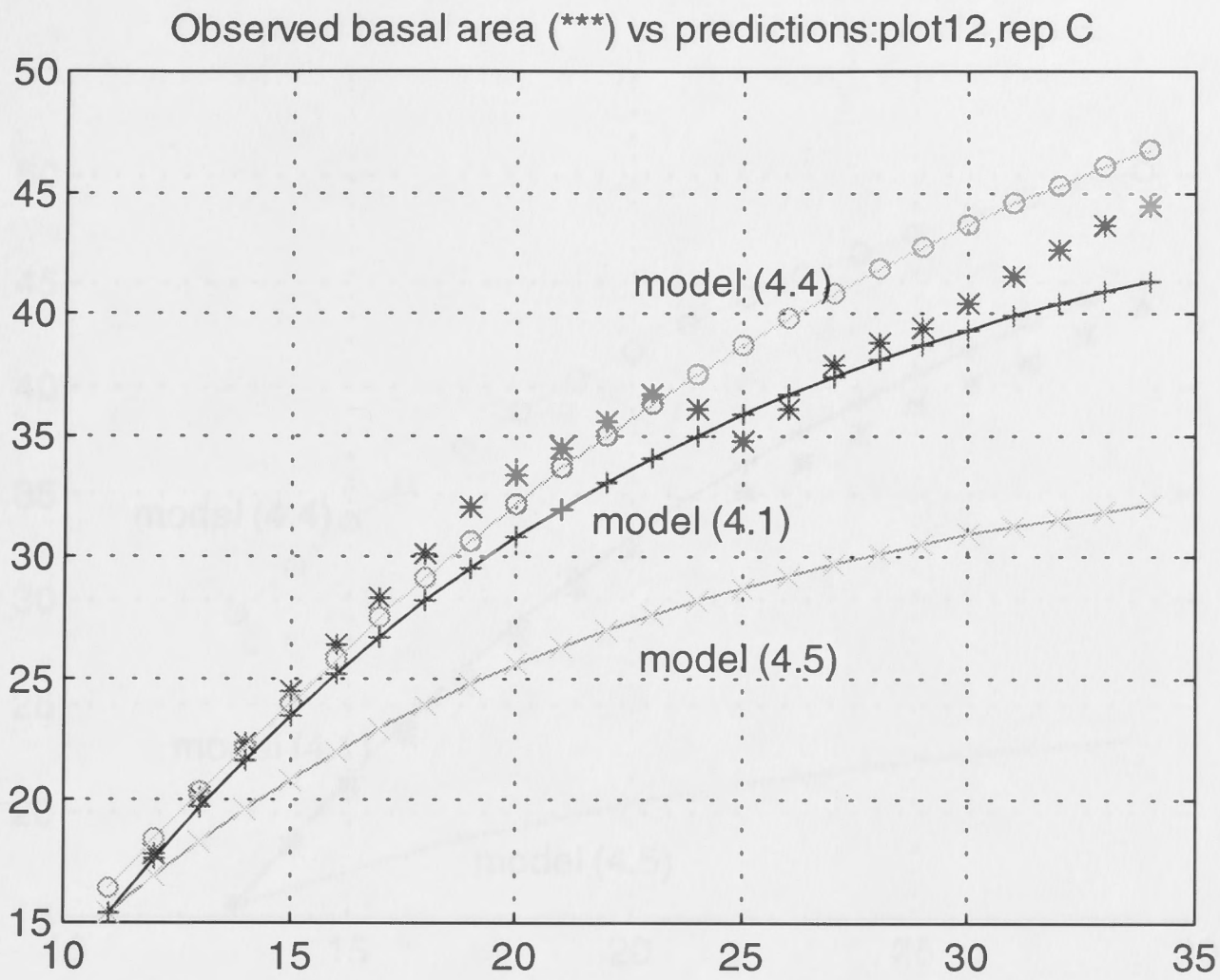


Figure 4.14: Thinning response at a residual density of 247 stems/ha at age 11.

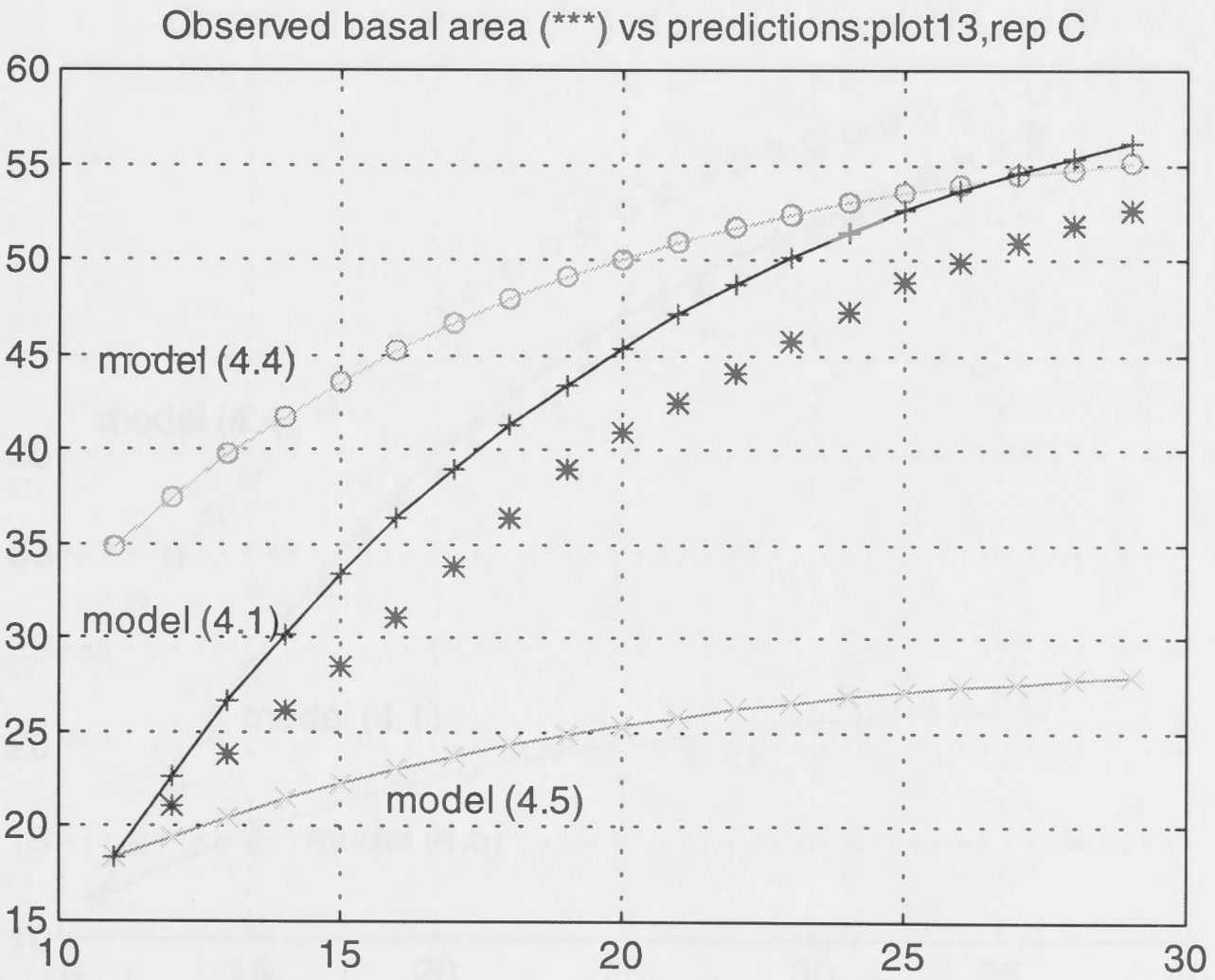


Figure 4.15: Thinning response at a residual density of 988 stems/ha at age 11.

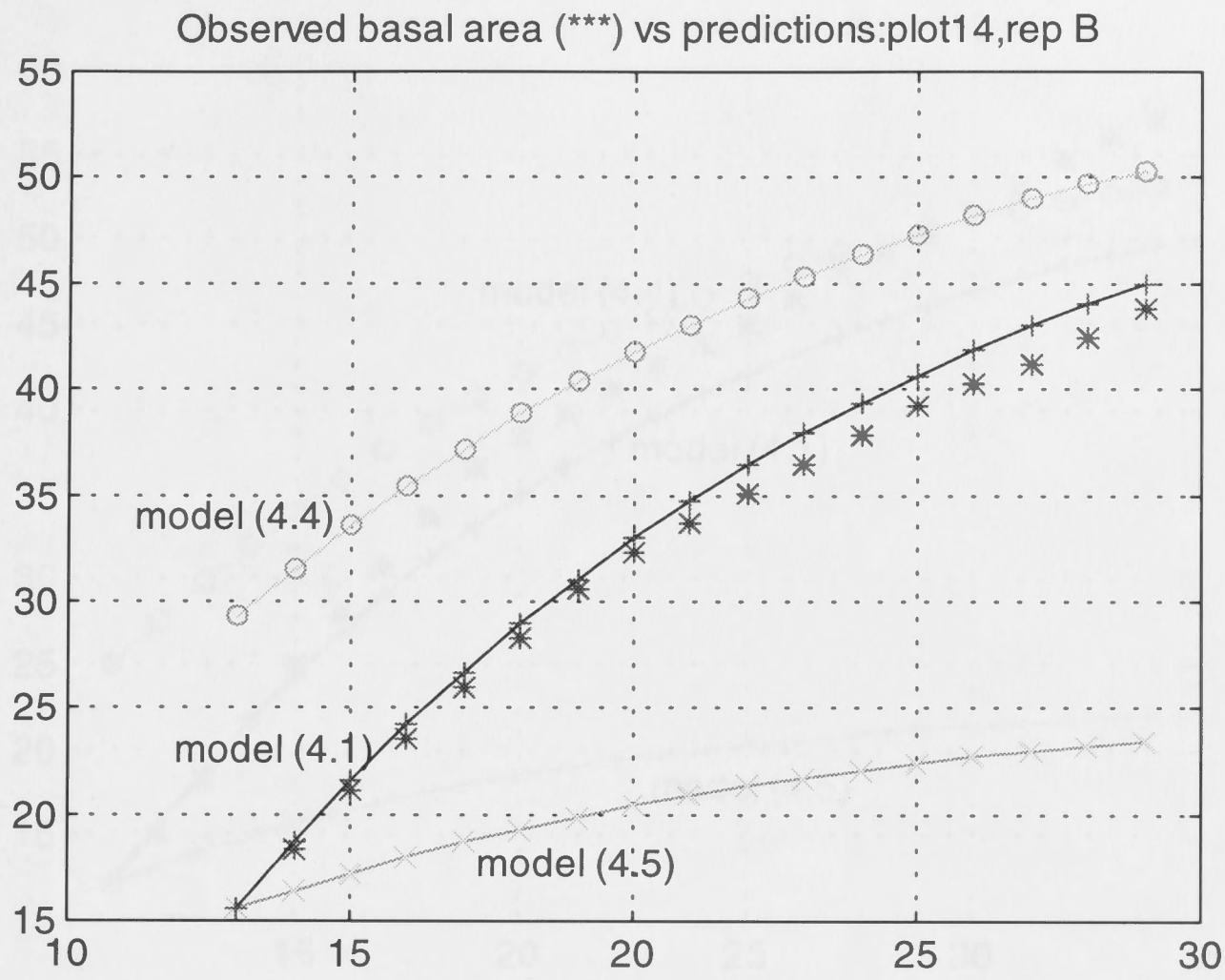


Figure 4.16: Thinning response at a residual density of 482 stems/ha at age 13.

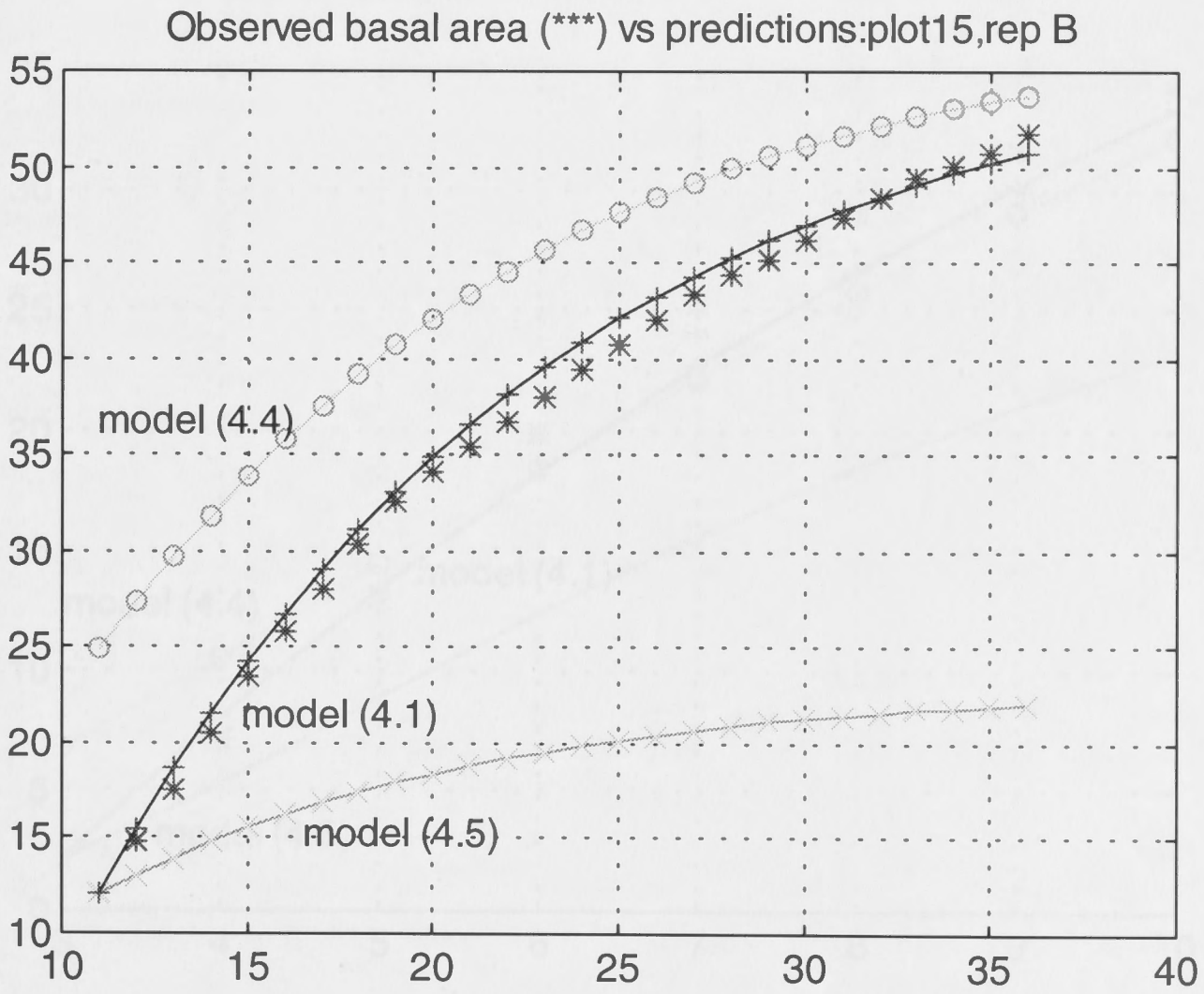


Figure 4.17: Thinning response at a residual density of 494 stems/ha at age 11.

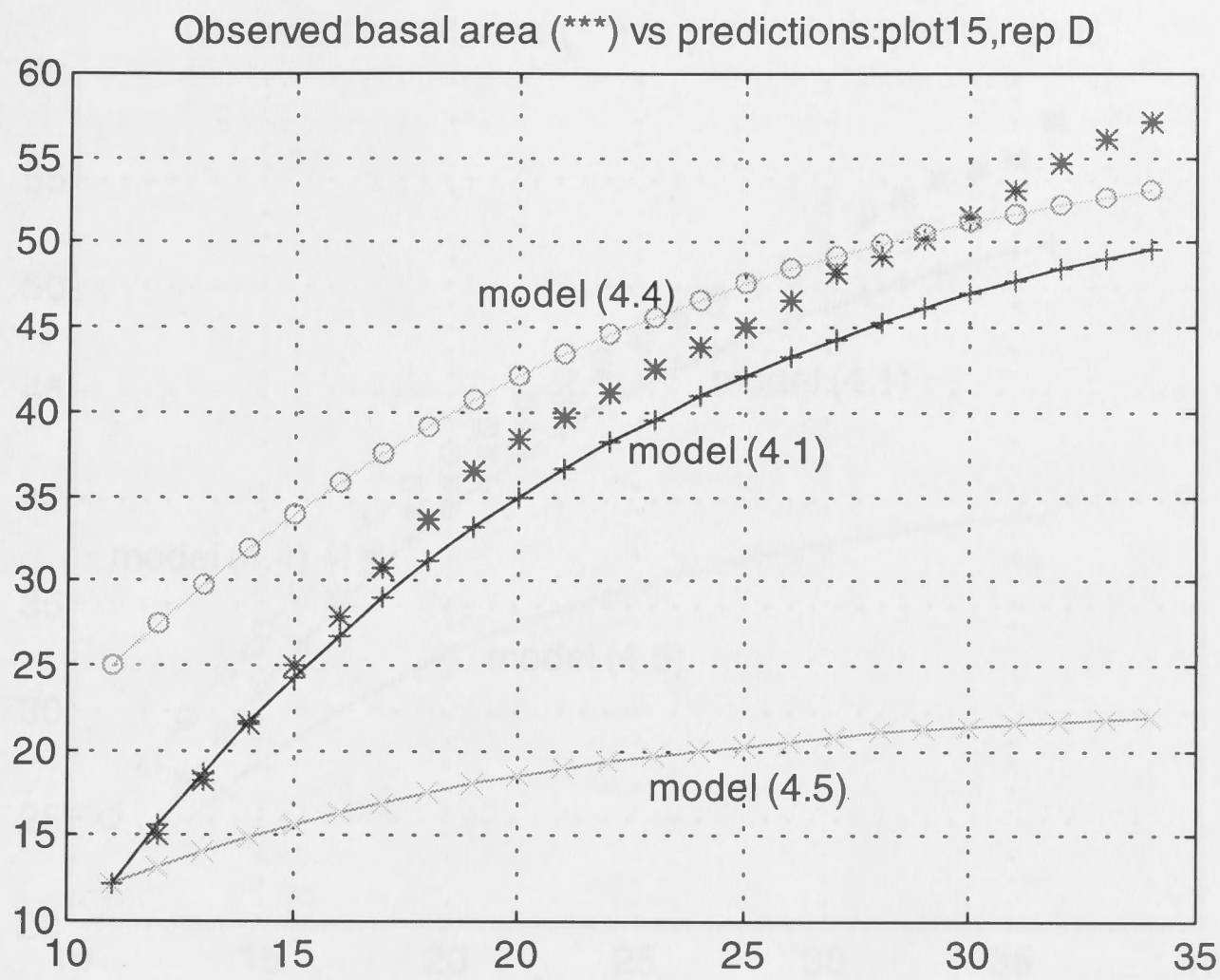


Figure 4.18: Thinning response at a residual density of 494 stems/ha at age 11.

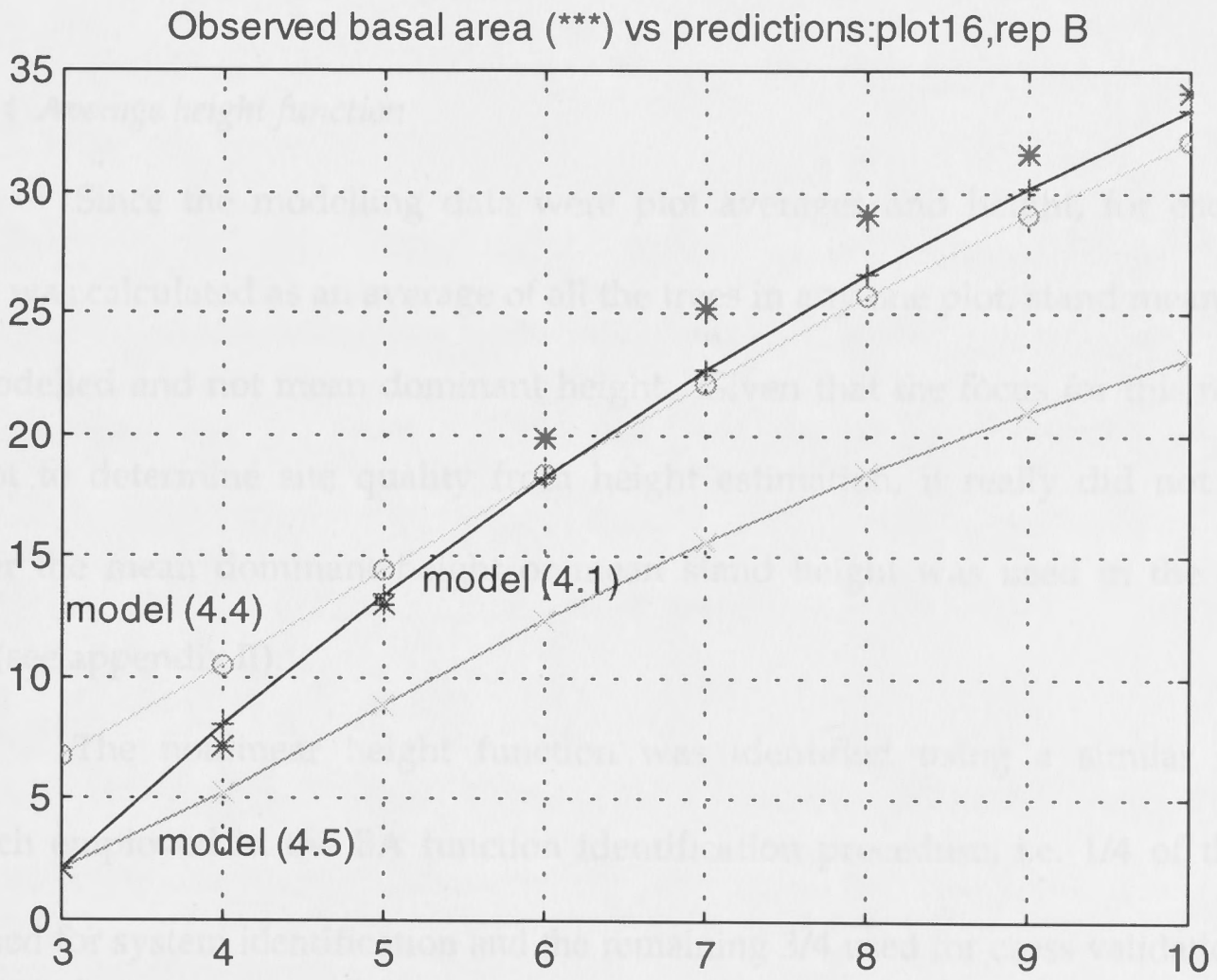


Figure 4.19: Thinning response at a residual density of 988 stems/ha at age 3.

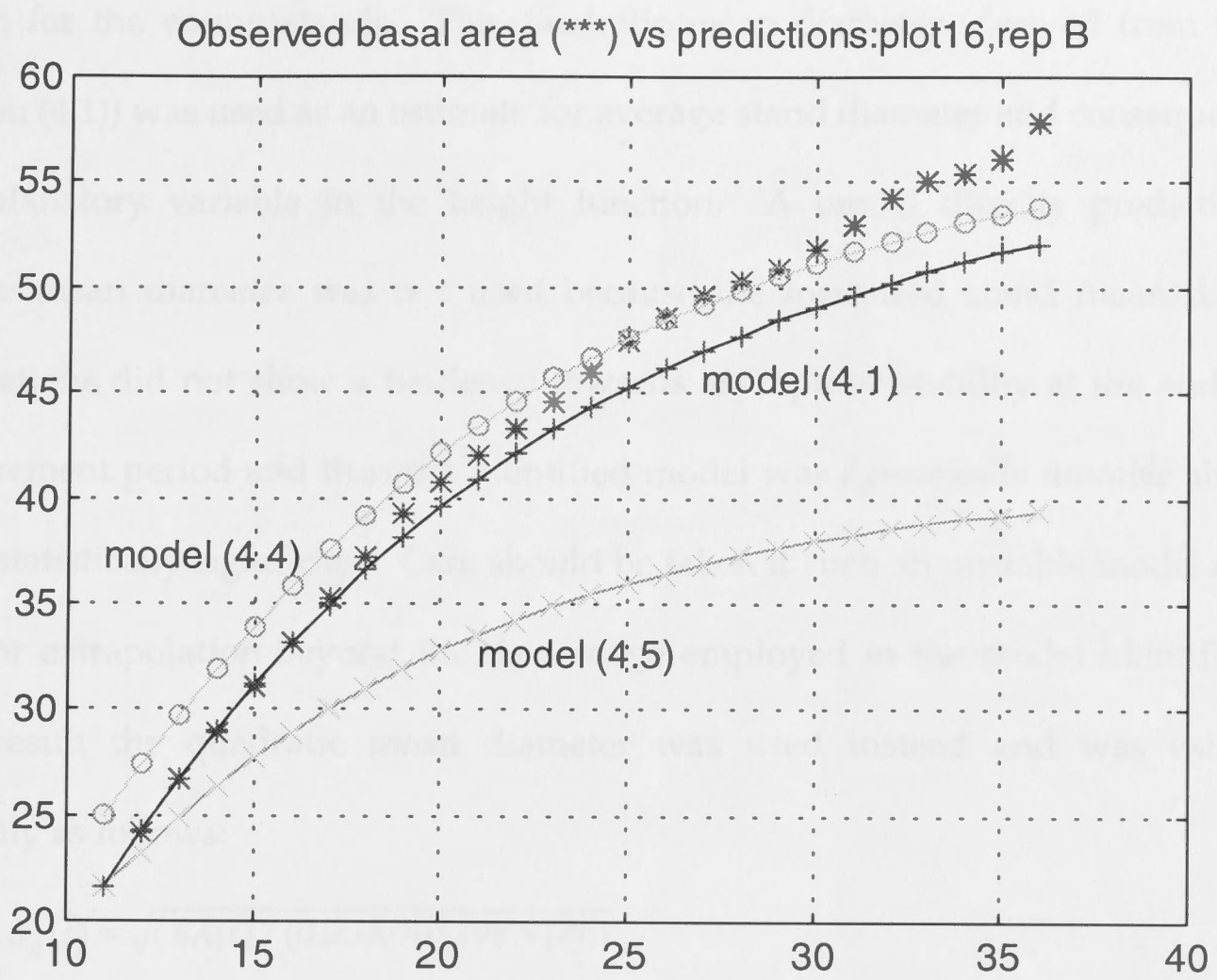


Figure 4.20: Thinning response at a residual density of 494 stems/ha at age 11.

4.6.4 Average height function

Since the modelling data were plot averages and height, for each time period, was calculated as an average of all the trees in any one plot, stand mean height was modelled and not mean dominant height. Given that the focus for this research was not to determine site quality from height estimation, it really did not matter whether the mean dominant height or mean stand height was used in the control model (see appendix II).

The nonlinear height function was identified using a similar 2-stage approach employed in the BA function identification procedure, i.e. 1/4 of the data were used for system identification and the remaining 3/4 used for cross validation. The height function was developed from plots 1-8 and the height growth response to density was such that the greater the stem number, the greater the rate of mean height

growth for the young stands. The quadratic mean diameter (derived from the BA function (4.1)) was used as an estimate for average stand diameter and consequently as an explanatory variable in the height function. A model directly predicting the average mean diameter was not used because the measured stand mean diameter observations did not show a tendency towards asymptotic stability at the end of the measurement period and thus the identified model was *dynamically unstable* although it was statistically significant. Care should be taken if such an unstable model is to be used for extrapolation beyond the data range employed in the model identification. As a result the quadratic mean diameter was used instead and was estimated indirectly as follows:

$$d_q(t) = \sqrt{(BA(t) / (0.0000785398\, N(t)))} \tag{4.6}$$

where

$d_q(t)$ = quadratic mean diameter (cm) at time t

$BA(t)$ = stand basal area (m^2/ha) at time t

$0.0000785398 = [(pi/4)/10\,000]$ m^2

$N(t)$ = stand density (stems/ha) at time t

The data (estimated quadratic mean diameter(input)/height(output)) showed stationarity and there were no delays in the height response. From this information, different model structures with different orders were identified (see appendix I). The orders of these models were tested to find the structure with the least number of parameters and yielding the best fit. The search was narrowed down to two models, an ARX(2 1 0) and an ARX(1 1 0). A further check on the zero-pole cancellation found the ARX(1 1 0) model adequate for describing the mean height. The height function is shown below:

$$H(t) = a\, H(t\text{-}1) + b\, d_q(t) \tag{4.7}$$

where

$H(t)$ = average stand height (m) at time t

$d_q(t)$ = estimated quadratic mean diameter (cm) at time t

$$a = 0.782; \text{ for } x \geq 1000 \text{ stems/ha} \tag{4.8}$$

$$= 0.85; \text{ for } 1000 > x \geq 400 \text{ stems/ha}$$

$$= 0.913; \text{ for } 400 > x \geq 124 \text{ stems/ha}$$

$$b = 0.19 + 0.03\frac{x}{1000}; \text{ for } x \geq 1000 \text{ stems/ha} \tag{4.9}$$

$$= 0.095 + 0.05\frac{x}{1000}; \text{ for } 1000 > x \geq 400 \text{ stems/ha}$$

$$= 0.035 + 0.1\frac{x}{1000}; \text{ for } 400 > x \geq 124 \text{ stems/ha}$$

The autocorrelation function of the residuals and cross correlation between the residuals and the estimated quadratic mean diameter for the height function (4.7), showed white noise properties within 99% confidence. More statistical validation of the eight height models that were developed from each of the first replicates in plots 1-8 is shown in Table 4.5 and the models have good statistical properties.

Table 4.5: Statistical validation of the eight mean stand height models identified from a single replicate from each of the eight plots (1-8).

PLOT REPLICATES (used for modelling)	*INITIAL DENSITY (stems/ha)*	*MEAN SQUARED ERROR*	*CORRELATION COEFFICIENT*	*RESIDUALS*
1A	2878 (at age 8)	0.87	0.99	horizontal band
2A	1458 (at age 8)	0.49	0.99	curvilinear band
3A	988	0.19	0.99	horizontal band
4A	744	0.73	0.99	horizontal band
5A	494	0.59	0.99	curvilinear band
6A	371	0.44	0.99	horizontal band
7A	247	0.65	0.99	horizontal band
8A	124	0.75	0.99	slightly decreasing

trend

Empirical evidence from thinning experiments indicates that for many commercially important species height growth is not greatly affected by the manipulation of stand density (Clutter et al., 1983). However, Menzies et al., (1989) found out that for *Pinus radiata* cuttings, there was a consistent trend of trees planted at 800 stems/ha being the tallest, followed by trees at stockings of 600 and 400 stems/ha, with trees from 200 stems/ha being the shortest. At age 6, trees from 800 and 600 stems/ha initial density were *significantly taller* than trees at 400 stems/ha, and trees at 200 stems/ha were *significantly shorter* than those for the 3 higher stocking levels. *P. patula* followed a similar trend and this relationship was found to be valid within the range of densities normally encountered in South African plantations (von Gadow, 1983).

Note that the observations by Menzies et al., (1989) were at young ages and *P. patula* data show that the height from all the different stand densities tend towards the same maximum value at the end of the measurement period. Thus different stand densities cause polymorphic behaviour of height growth but the height will tend towards the same asymptotic maximum on one site. How these differences in height growth due to different stand densities affect the accuracy of volume predictions, would be an interesting area of research.

When plotted against initial stand levels, parameters a and b showed trajectories that were discontinuous at certain densities i.e. piecewise continuous. At each point of discontinuity a significant 'step change' in the value of a or b was observed with the result of three separate trends (within each piecewise continuous trajectory) being recognised. It was not possible to accurately determine the exact initial densities at these points of discontinuity and therefore the estimations may be coarse. Each trend of a piecewise continuous trajectory was represented by a constant or polynomial function for both a and b values.

Finally cross validations done were quite good and eight of those, each from plots 1-8 are in Figures 4.21-28. Table 4.6 shows the mean squared errors of these validated plots. The correlations functions are shown in appendix 4B.

Table 4.6: Mean squared errors from cross validation with height model (4.7)

PLOT REPLICATES (not used in model development)	*INITIAL DENSITY (stems/ha)*	*MEAN SQUARED ERROR (for height predictions)*
1B	2889	0.46
2B	1470	0.84
3C	988	0.68
4D	741	0.47
5D	494	0.47
6B	371	0.72
7C	247	0.47
8C	124	0.60

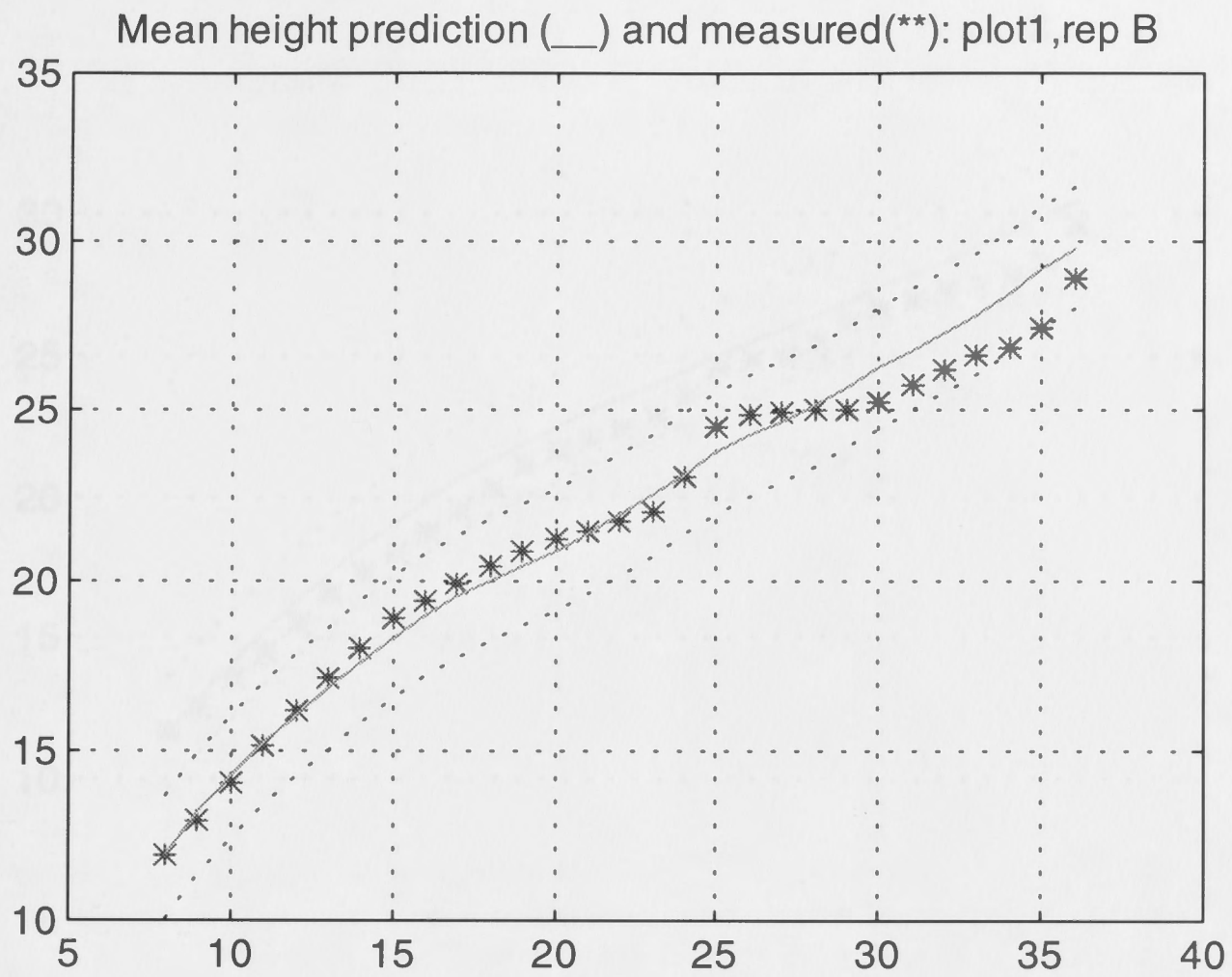


Figure 4.21: Cross validation of function (4.7) where initial density was 2953 stems/ha.

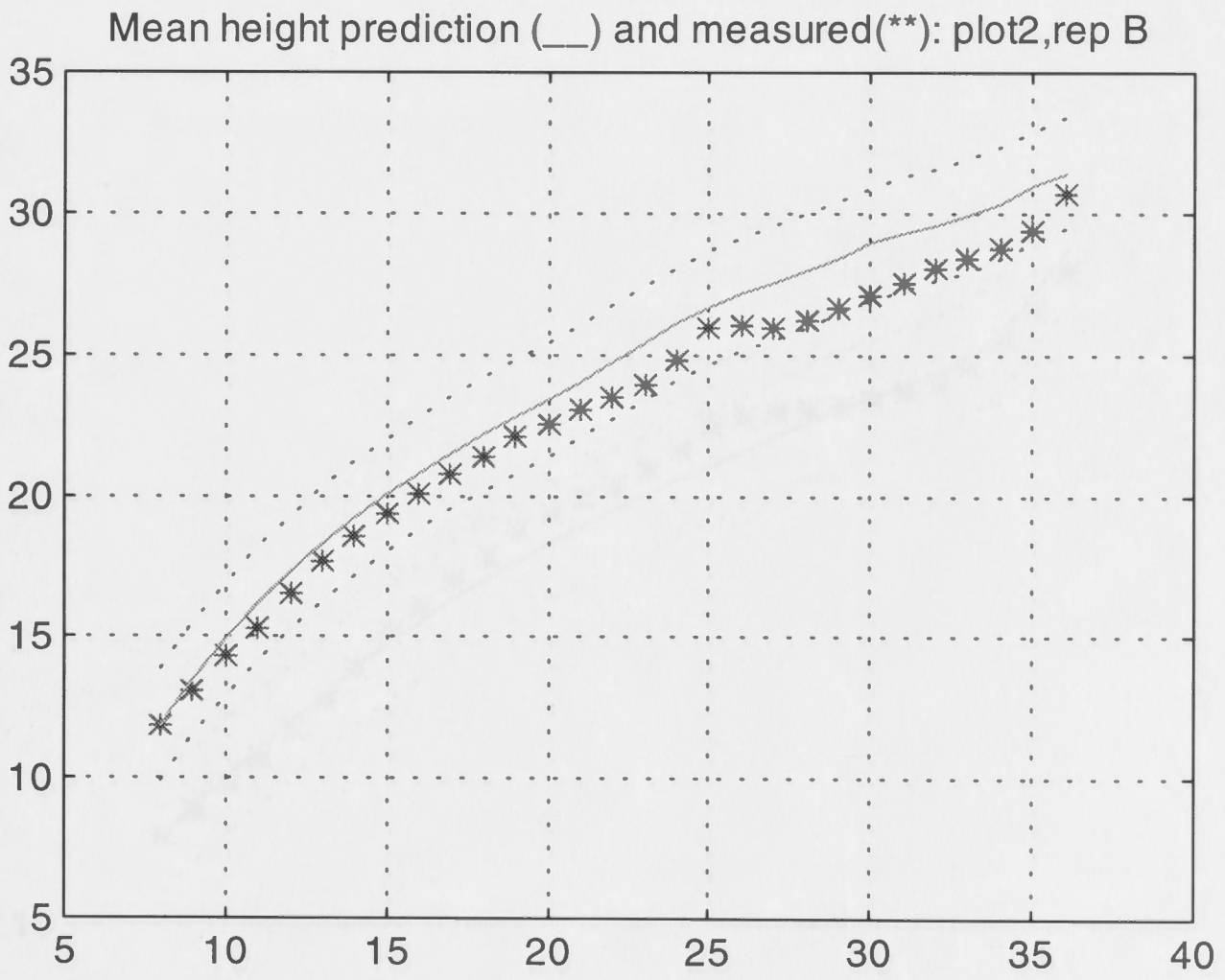


Figure 4.22: Cross validation of function (4.7) where initial density was 1483 stems/ha

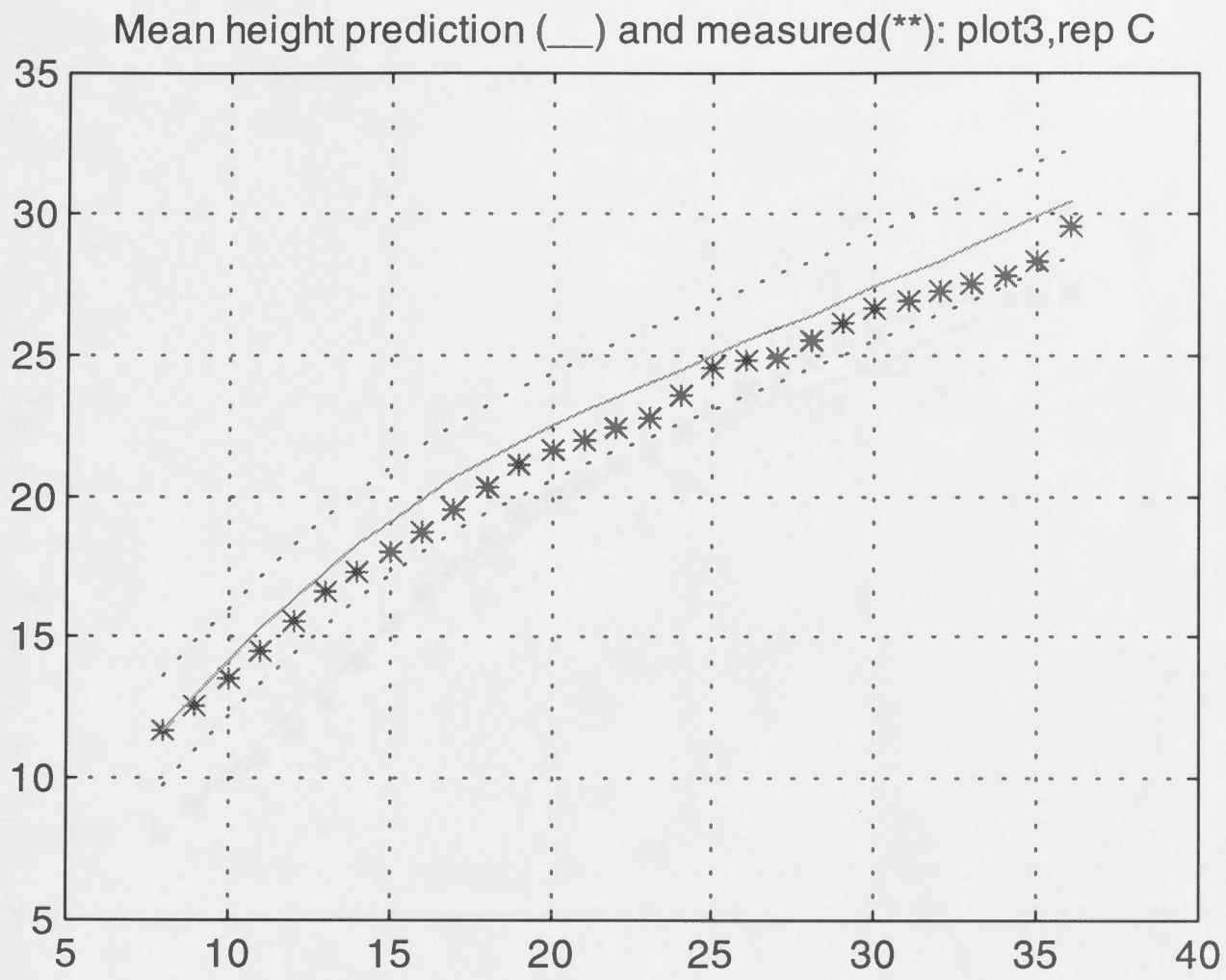


Figure 4.23: Cross validation of function (4.7) where initial density was 988 stems/ha

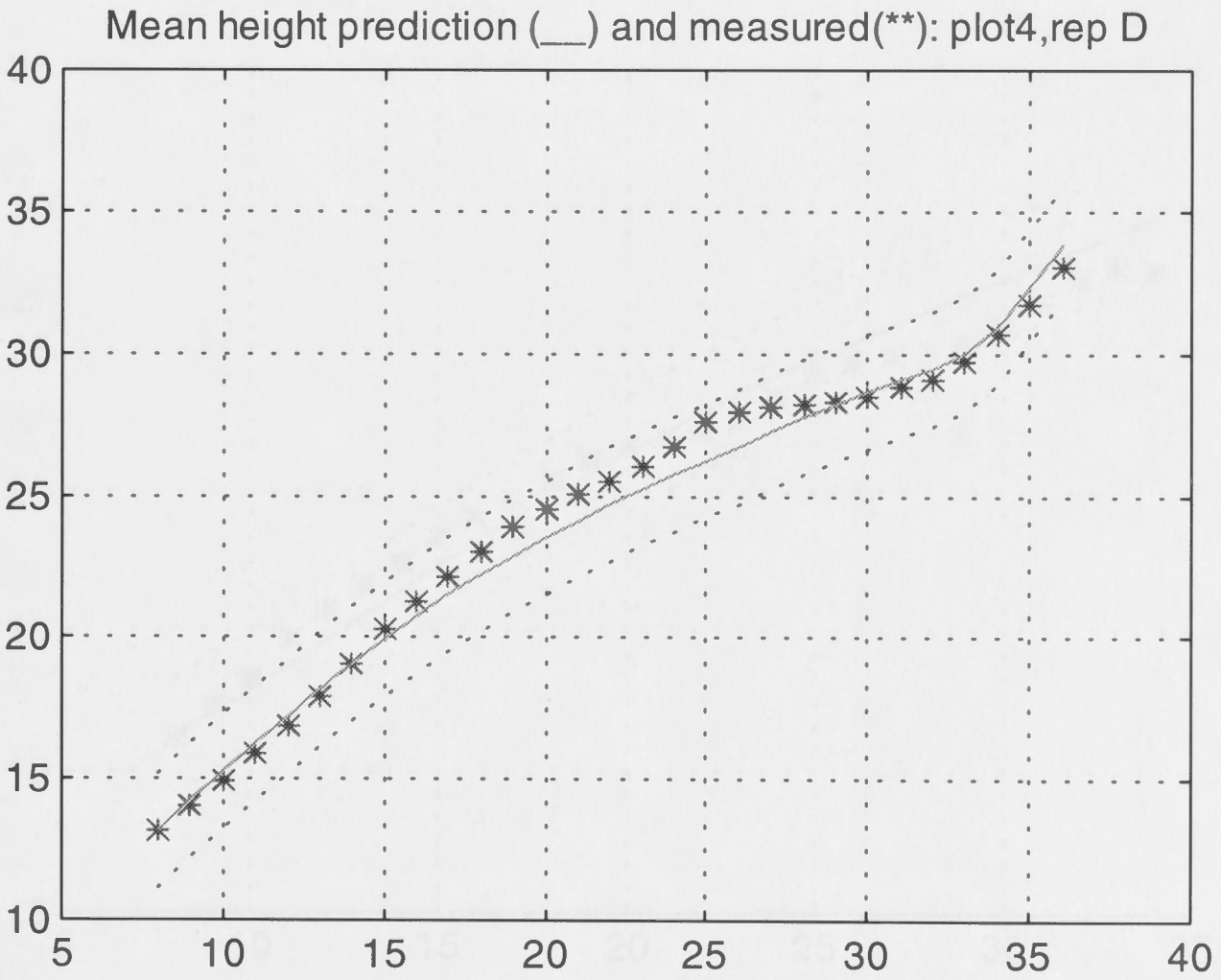


Figure 4.24: Cross validation of function (4.7) where initial density was 741 stems/ha

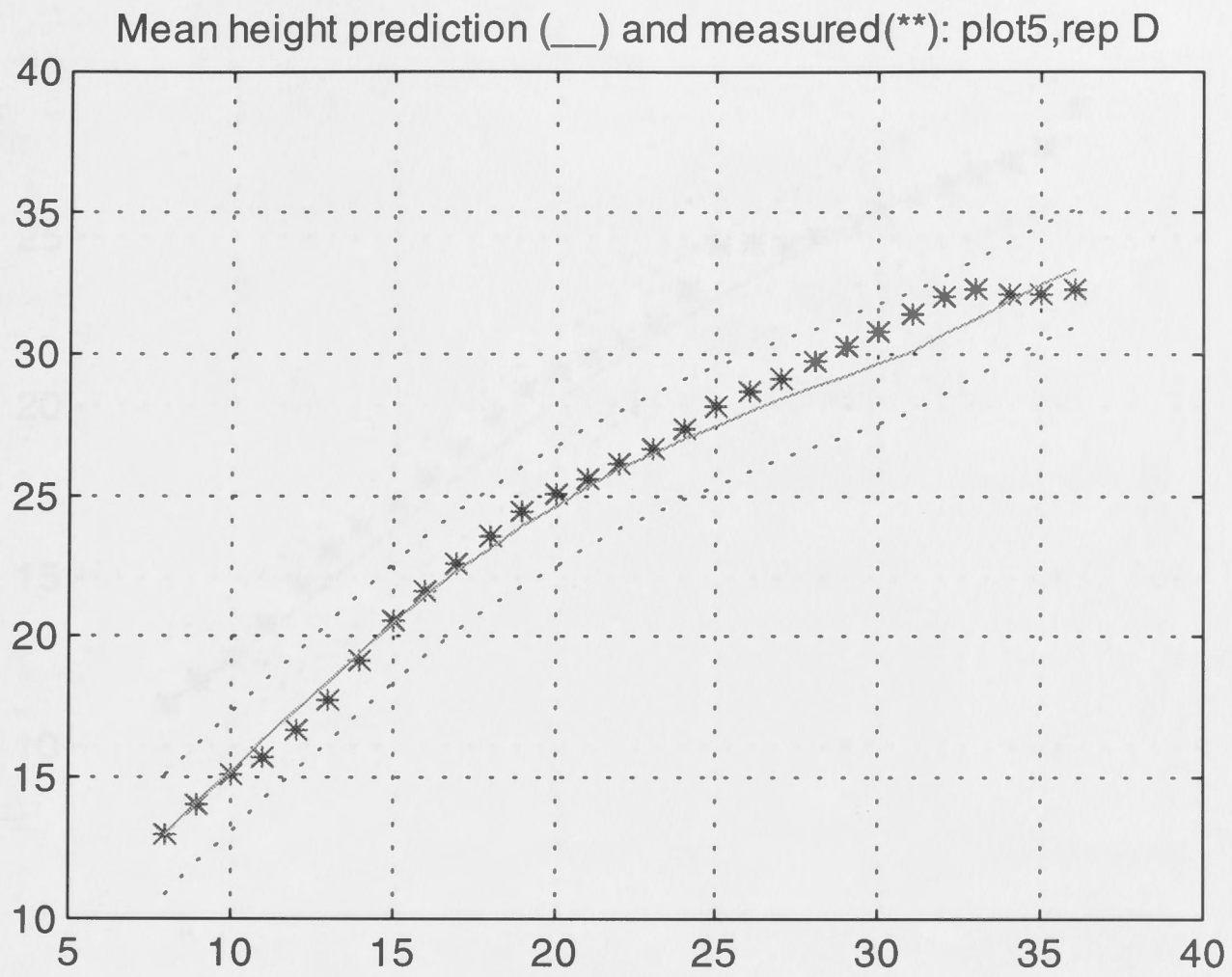


Figure 4.25: Cross validation of function (4.7) where initial density was 494 stems/ha

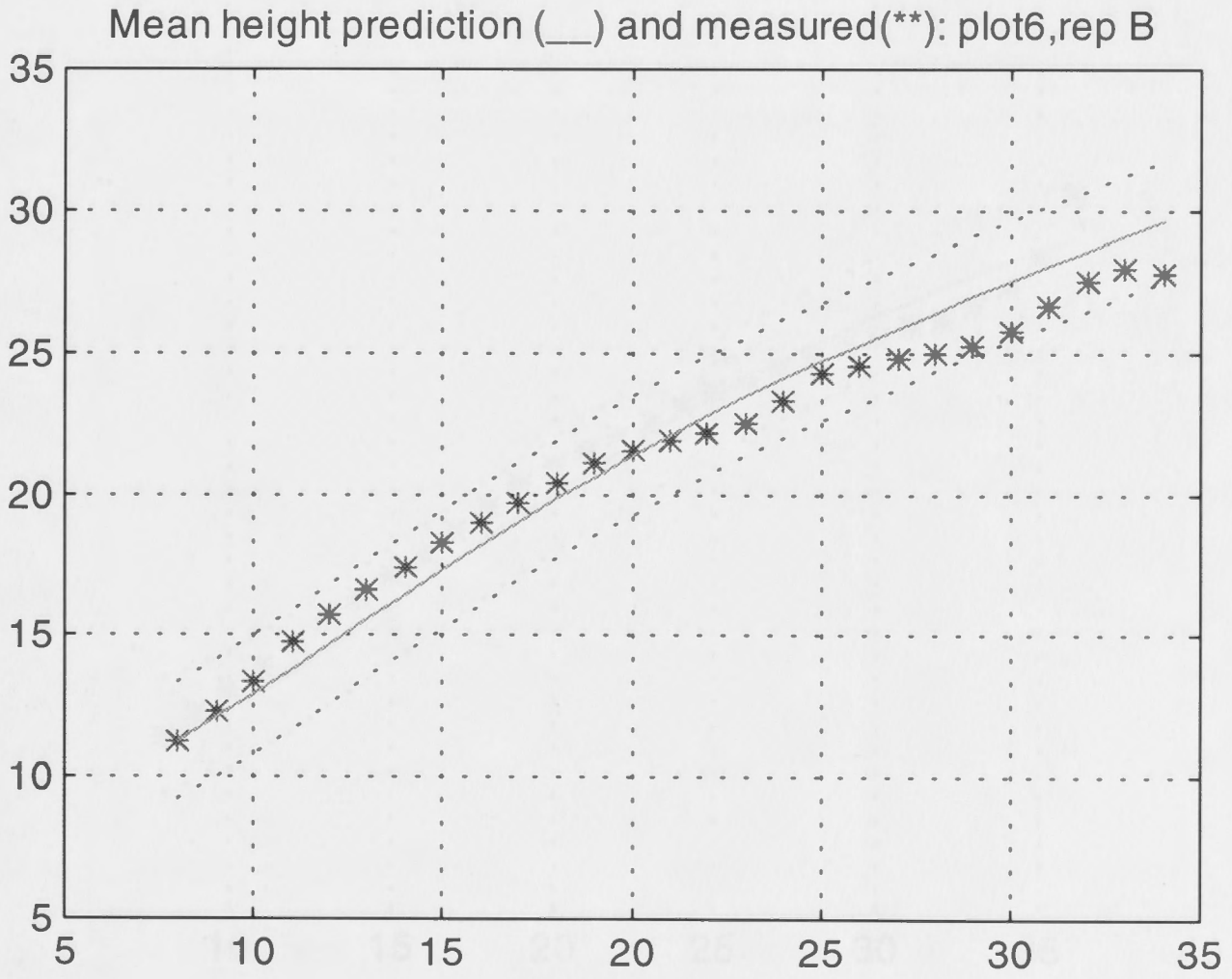


Figure 4.26: Cross validation of function (4.7) where initial density was 371 stems/ha

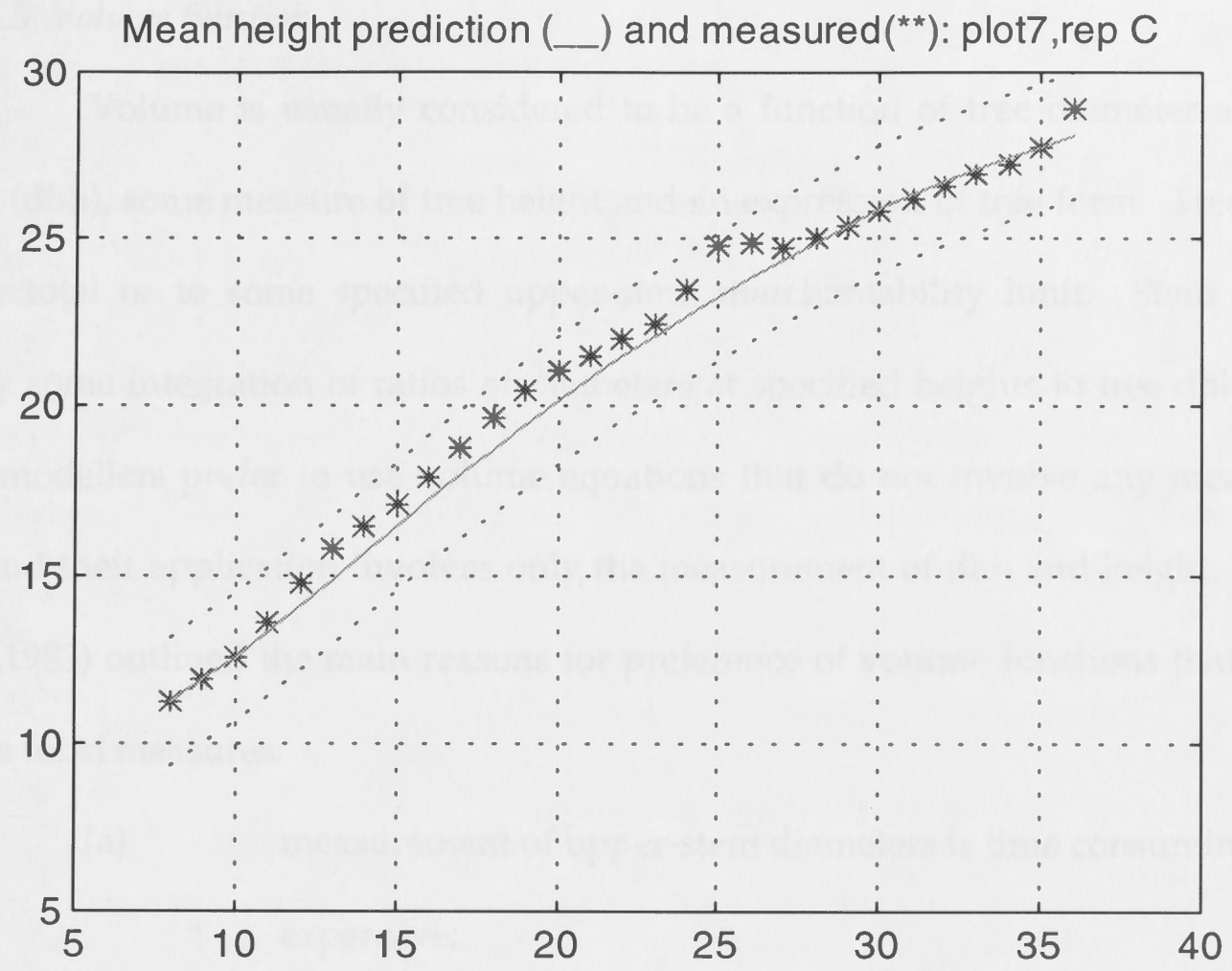


Figure 4.27: Cross validation of function (4.7) where initial density was 241 stems/ha

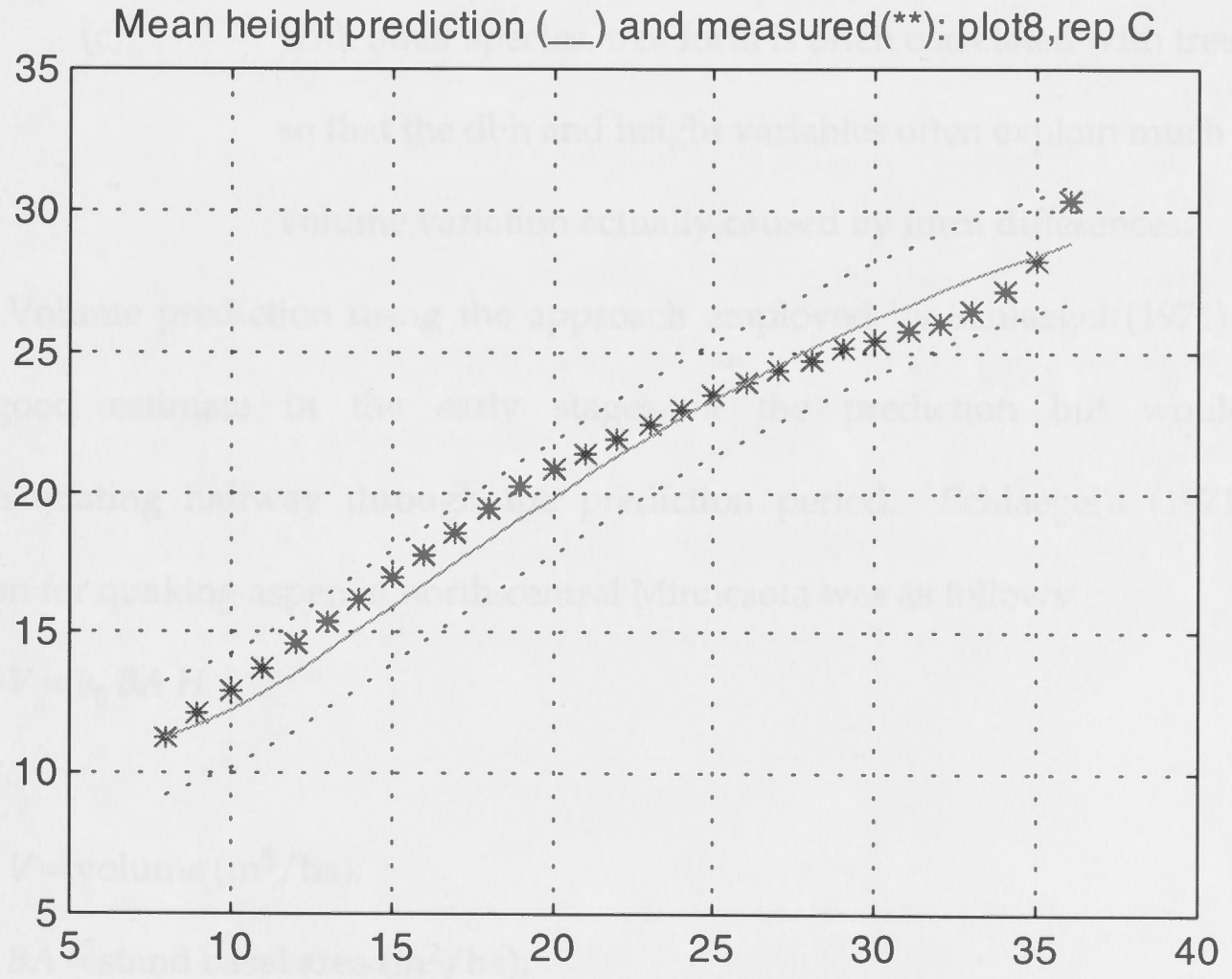


Figure 4.28: Cross validation of equation (4.7) where initial density was 124 stems/ha

4.6.5 Volume function

Volume is usually considered to be a function of tree diameter at breast height (dbh), some measure of tree height and an expression of tree form. Tree height can be total or to some specified upper-stem merchantability limit. Stem form is usually some integration of ratios of diameters at specified heights to tree dbh. Most forest modellers prefer to use volume equations that do not involve any measure of form and their application involves only the measurement of dbh and height. Clutter et al., (1983) outlined the main reasons for preference of volume functions that do not involve form measures:

(a) measurement of upper-stem diameters is time consuming and expensive;

(b) variation in tree form has a much smaller impact on tree volume than height or dbh variation; and

(c) with other species, tree form is often correlated with tree size, so that the dbh and height variables often explain much of the volume variation actually caused by form differences.

Volume prediction using the approach employed by Schlaegel (1971) gave a very good estimate in the early stages of the prediction but would start underestimating halfway through the prediction period. Schlaegel's (1971) yield equation for quaking aspen in north-central Minnesota was as follows:

$$V = b_0 BA\, H \tag{4.10}$$

where

V = volume (m^3/ha);

BA = stand basal area (m^2/ha);

H = average height of dominants and codominants (m);

$b_0 = 0.41898$.

A possible explanation for the underestimation is that the shape parameter, b_0, in equation (4.10) varies with time, from a near cone in young trees to a near cylinder in the older and larger trees. This natural tendency for tree form to become more cylindrical with age was studied by Stoate (1942), who showed that the increase in cylindricity with age was due to a relatively greater increase in height growth than diameter growth. In other words, the decrease in stem taper with age is primarily an additive effect of continued height growth. Open-grown trees, on the other hand, tend to overcome the influence of age and retain their strongly tapering stems until late in life provided their crowns remain vigorous (Larson, 1963).

To bias an equation of the form (4.10) to the shape of the older trees, only the top half of the data were used to estimate the shape parameter. This made the function fit well for all the data range. The equation of the form (4.10) was ideal for the young stands possibly for the following reasons:

(a) the equation was constrained to converge to zero at time zero; and

(b) the volume of small trees is insignificant in total plot volume.

However, to develop a single tree volume function (which, however, was not necessary for this control design) with a mathematical structure of equation (4.10), it would be necessary to make the shape parameter time-dependent. Using least squares estimation, the following stand volume equation was obtained for *P. patula* from plots 1-5 for replicates *A*, *B* and *C*:

$$V = b_1 BA\, H \tag{4.11}$$

where

V = stand volume over bark (m^3/ha);

BA = stand basal area estimated from equation (4.1), m^2/ha;

H = average stand height estimated from equation (4.7), m;

$b_1 = 0.4047$.

The above equation had a linear correlation coefficient (R) of *0.8823*. Although the R value was highly significant, it should be noted that the statistic is ignorant of the individual distributions of the samples under analysis (Press et al., 1992). The mean squared error (MSE) takes into account the individual distributions by measuring the expectation of the squared difference between the estimator and the parameter. In this case, the MSE [*10.68*], was significantly smaller than the variance [*7844.5*] of the estimated volume; this means that the function (4.11) is efficient in its estimations although it might be biased. Cross validation was done for equation (4.11) and the tests were fairly good (see Table 4.7 and Figures 4.29-36).

Table 4.5: Mean squared errors from cross validation with volume function (4.10).

PLOT REPLICATES *(not used in the modelling)*	*INITIAL DENSITY* *(stems/ha)*	*VARIANCE* *(of volume observations)*	*MEAN SQUARED ERROR* *(of predicted volume)*
1D	2792	30189	21.12
2D	1470	42858	28.76
3D	988	73349	56.88
4D	741	44666	45.66
5D	494	48649	26.93
6A	371	26008	28.08
7A	247	15811	17.99
8C	124	16323	15.27

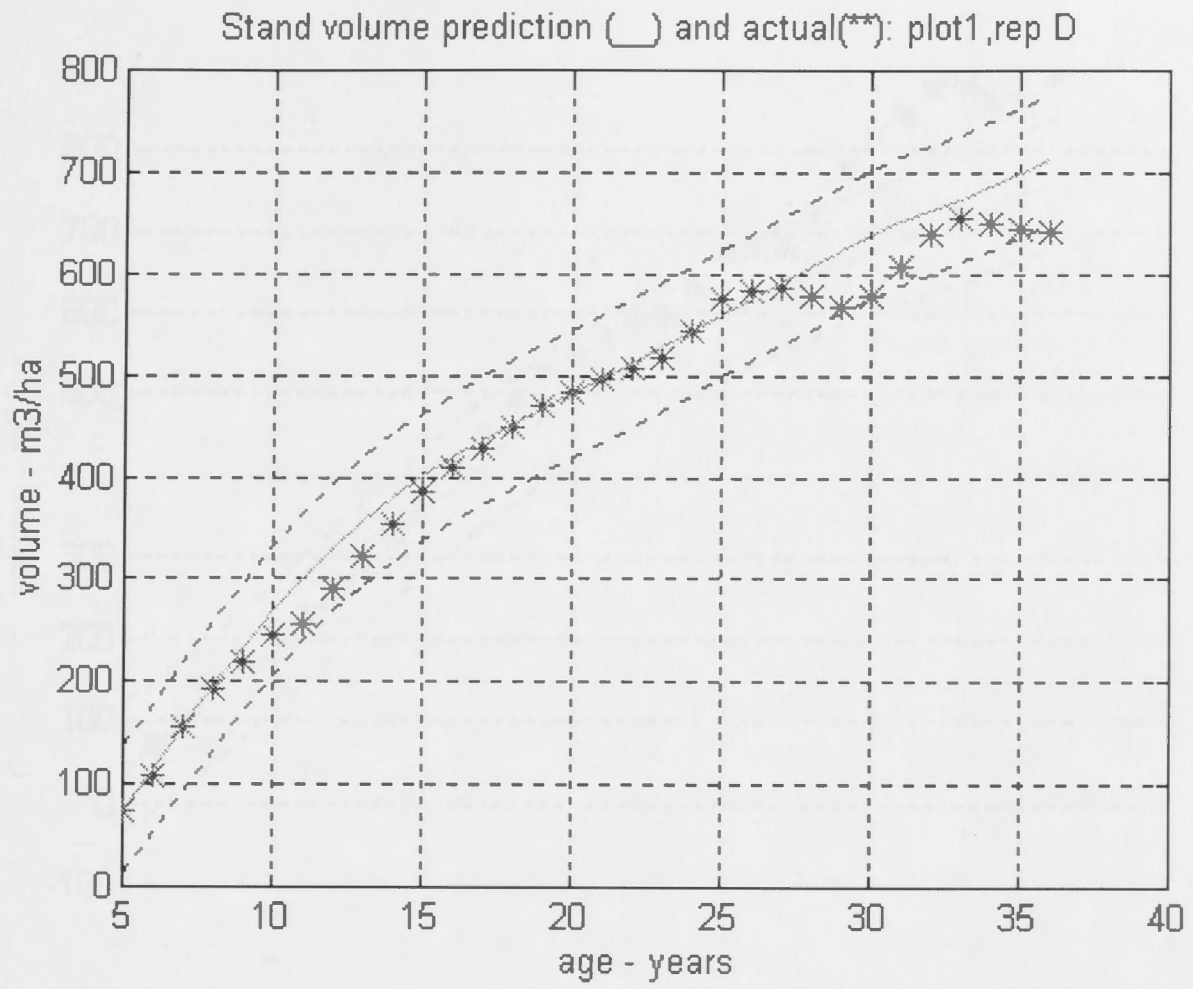


Figure 4.29: Cross validation of volume function (4.11) against observed volume

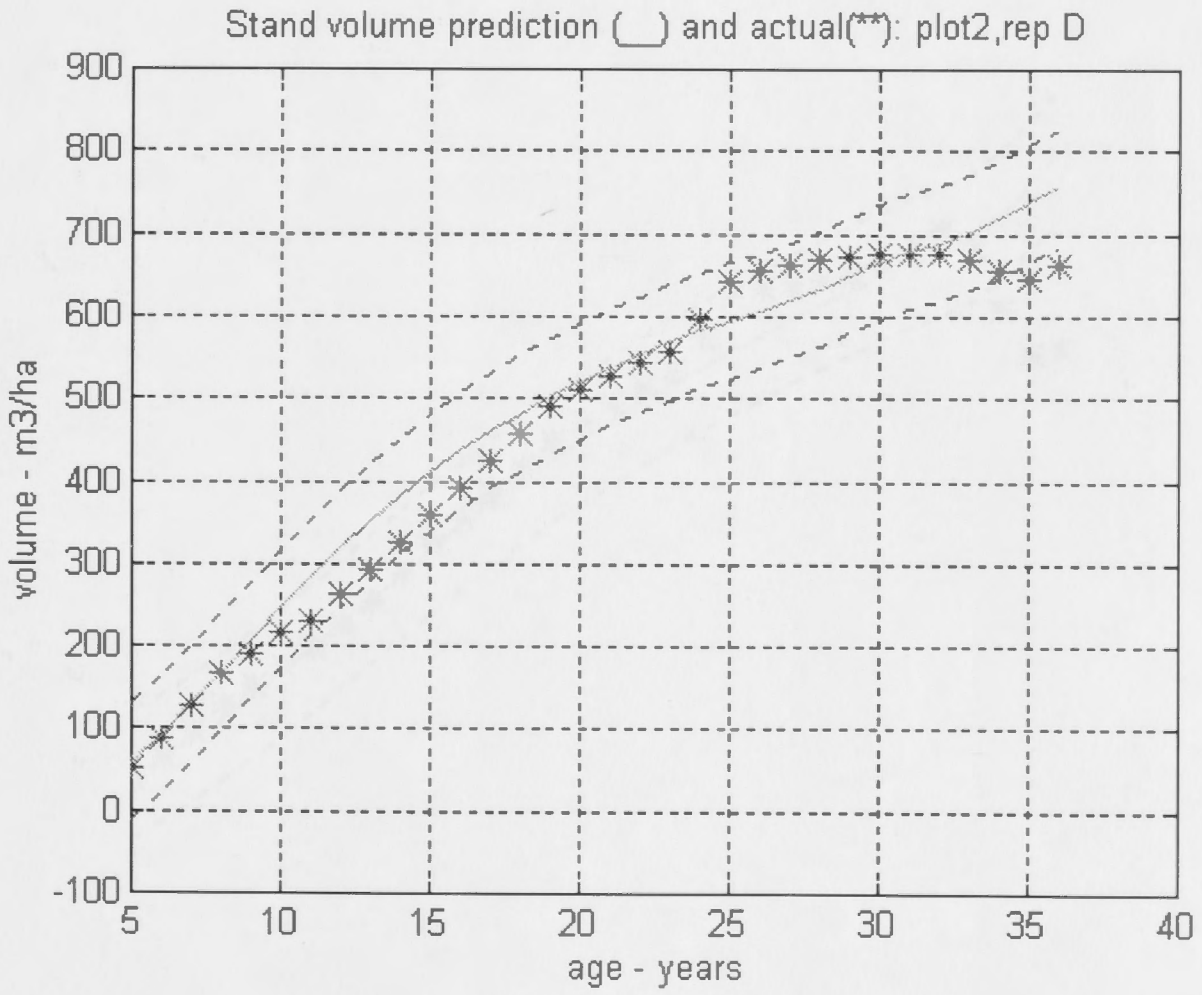


Figure 4.30: Cross validation of volume function (4.11) against observed volume

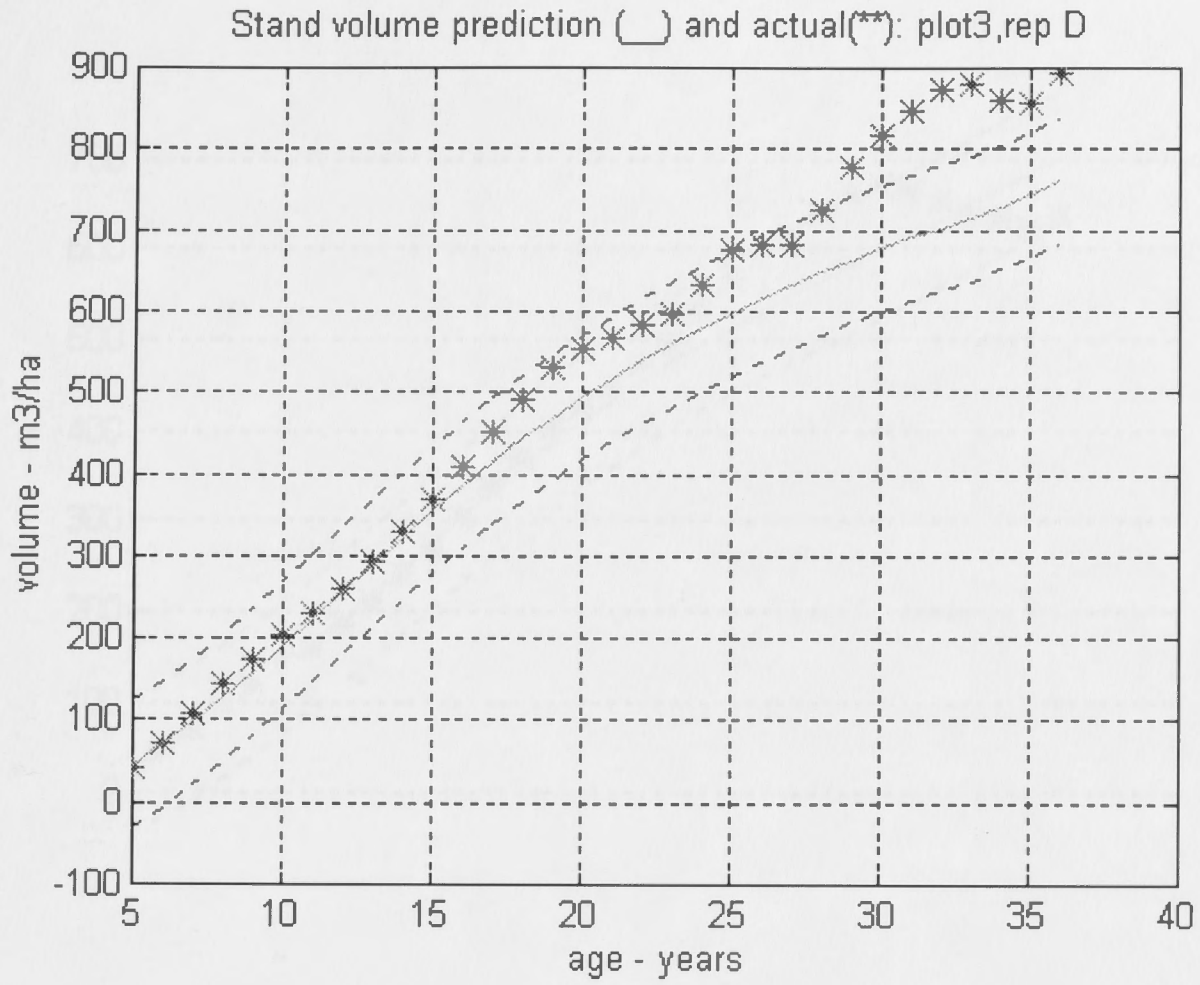


Figure 4.31: Cross validation of volume function (4.11) against observed volume

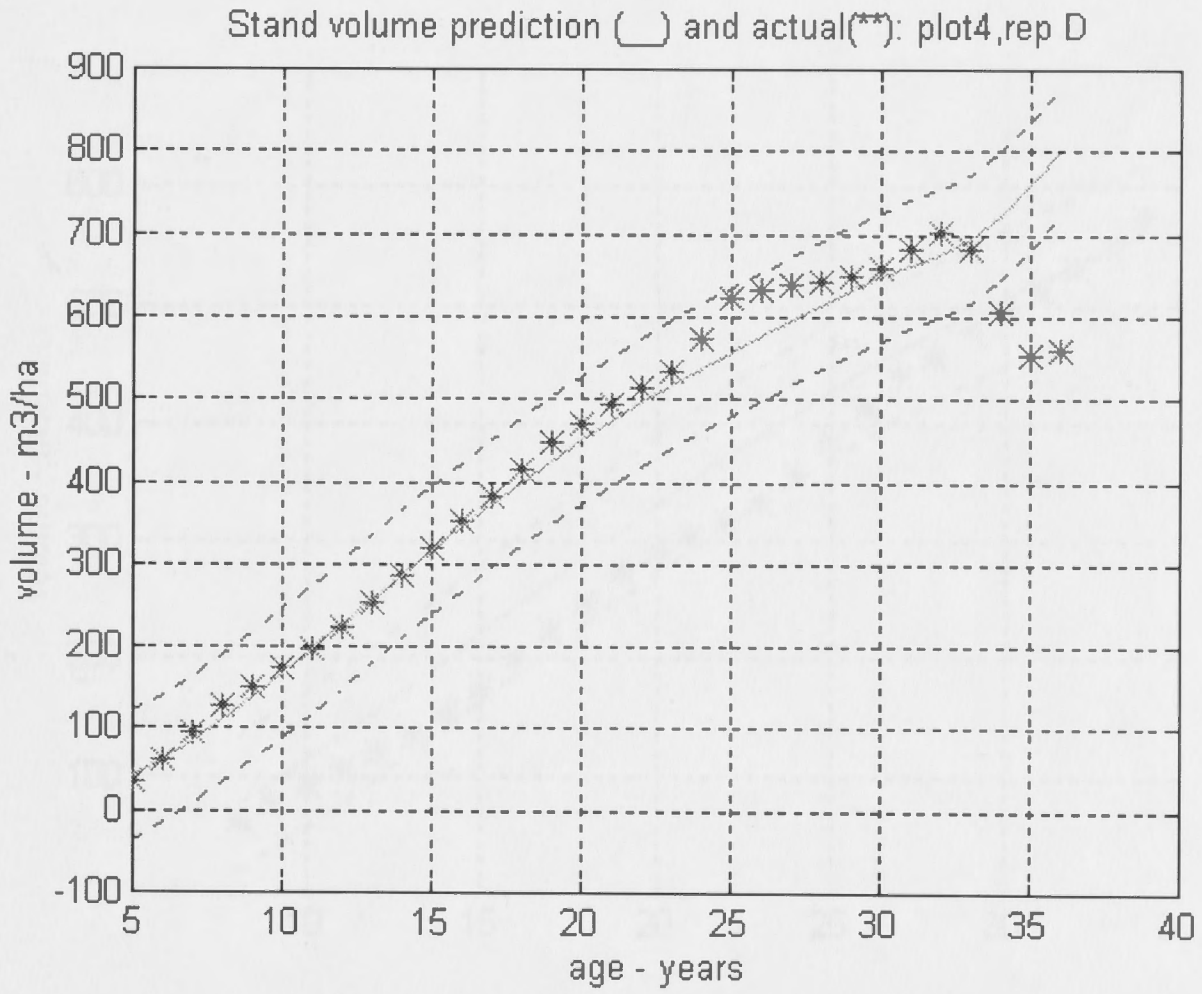


Figure 4.32: Cross validation of volume function (4.11) against observed volume

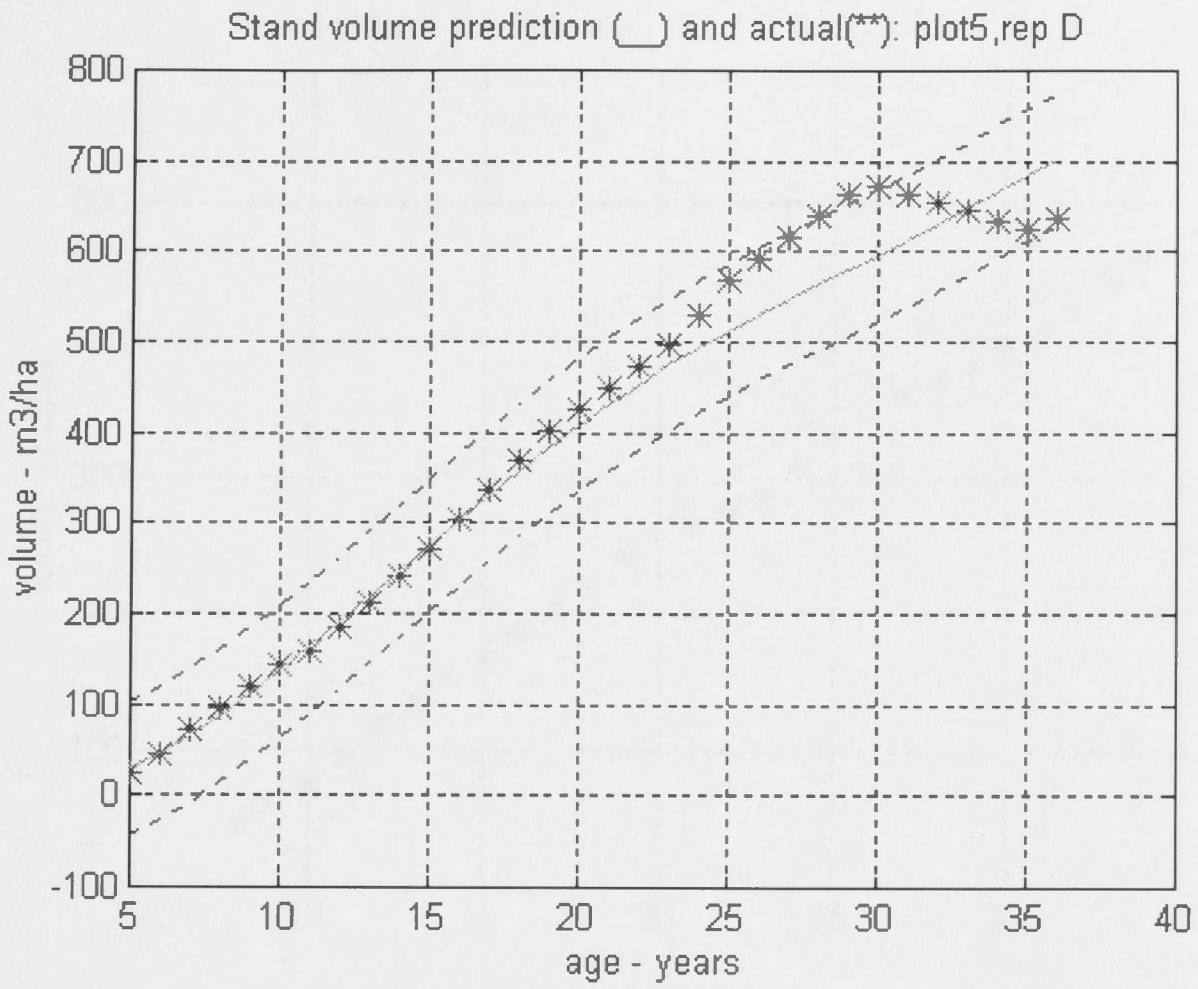


Figure 4.33: Cross validation of volume function (4.11) against observed volume

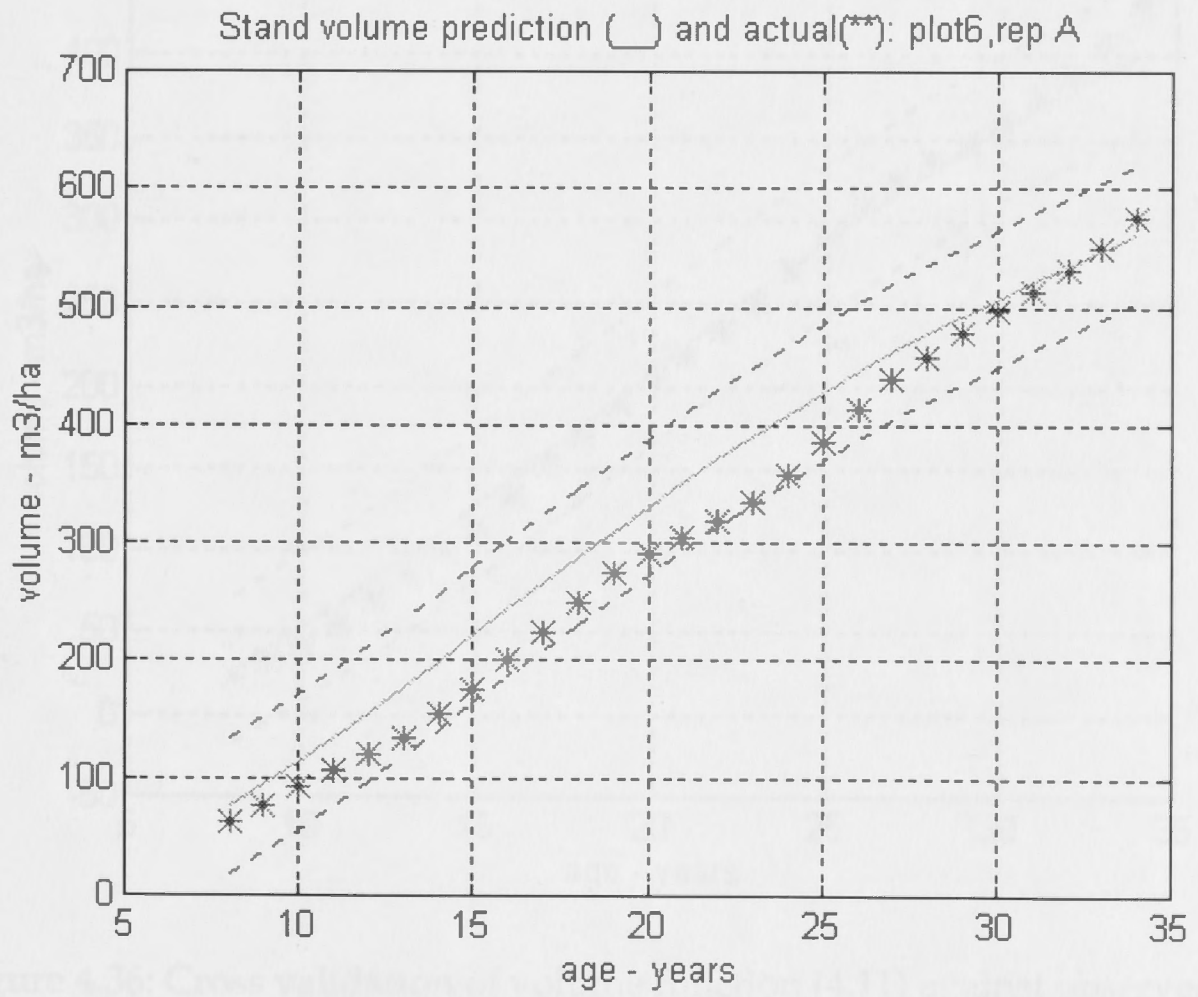


Figure 4.34: Cross validation of volume function (4.11) against observed volume

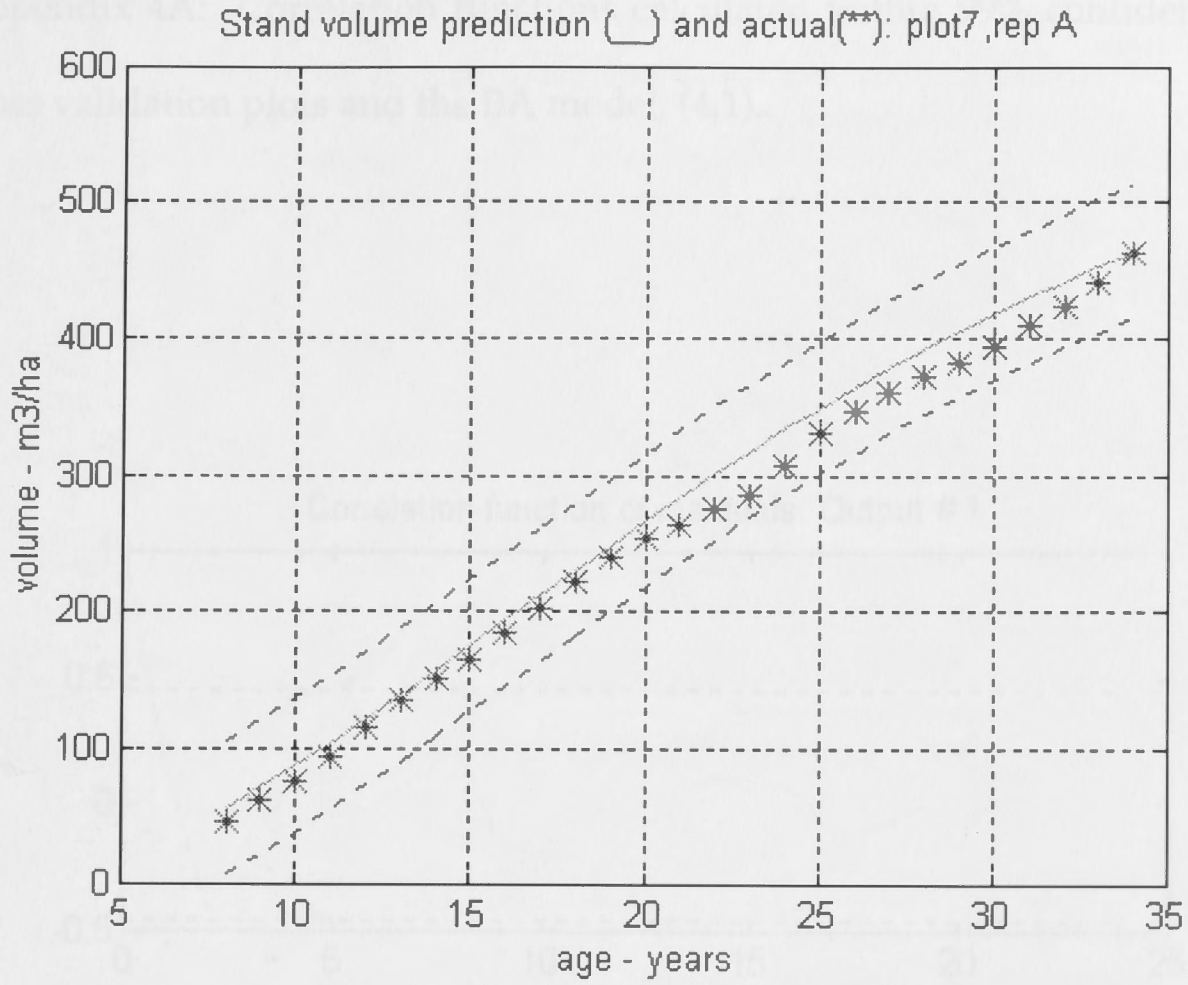


Figure 4.35: Cross validation of volume function (4.11) against observed volume

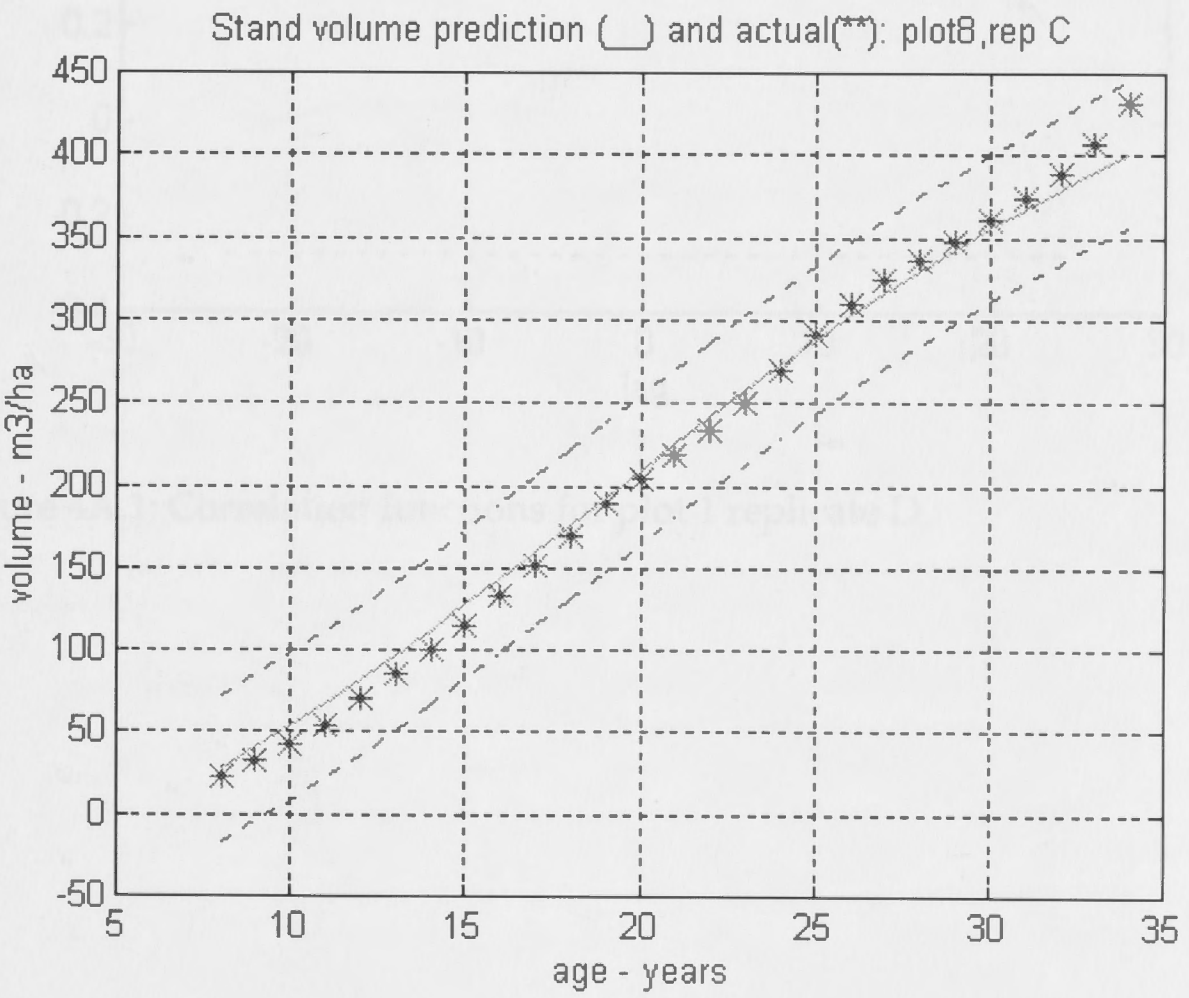


Figure 4.36: Cross validation of volume function (4.11) against observed volume

Appendix 4A: Correlation functions calculated within 99% confidence limits for the cross validation plots and the BA model, (4.1).

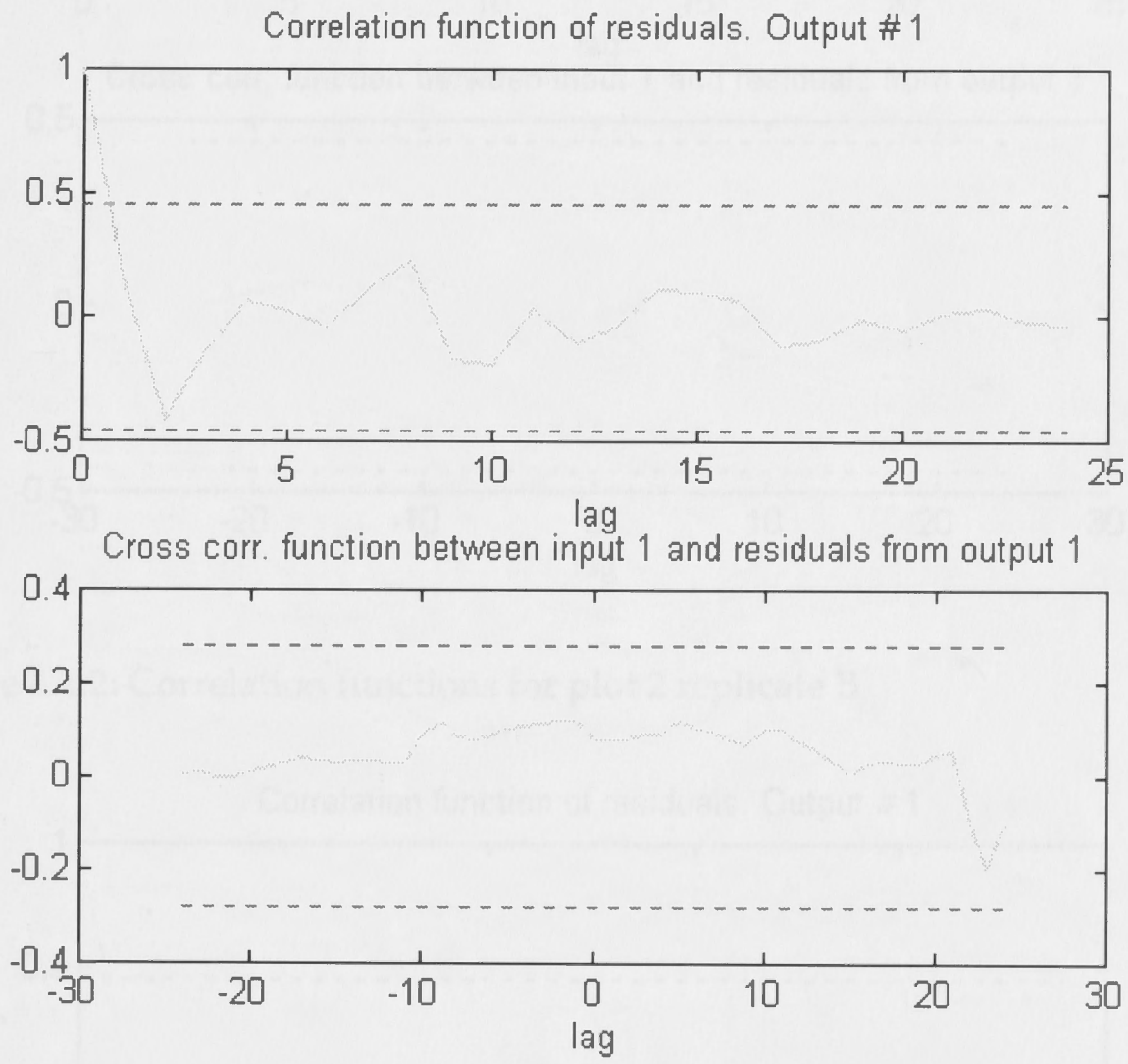


Figure 4A.1: Correlation functions for plot 1 replicate D

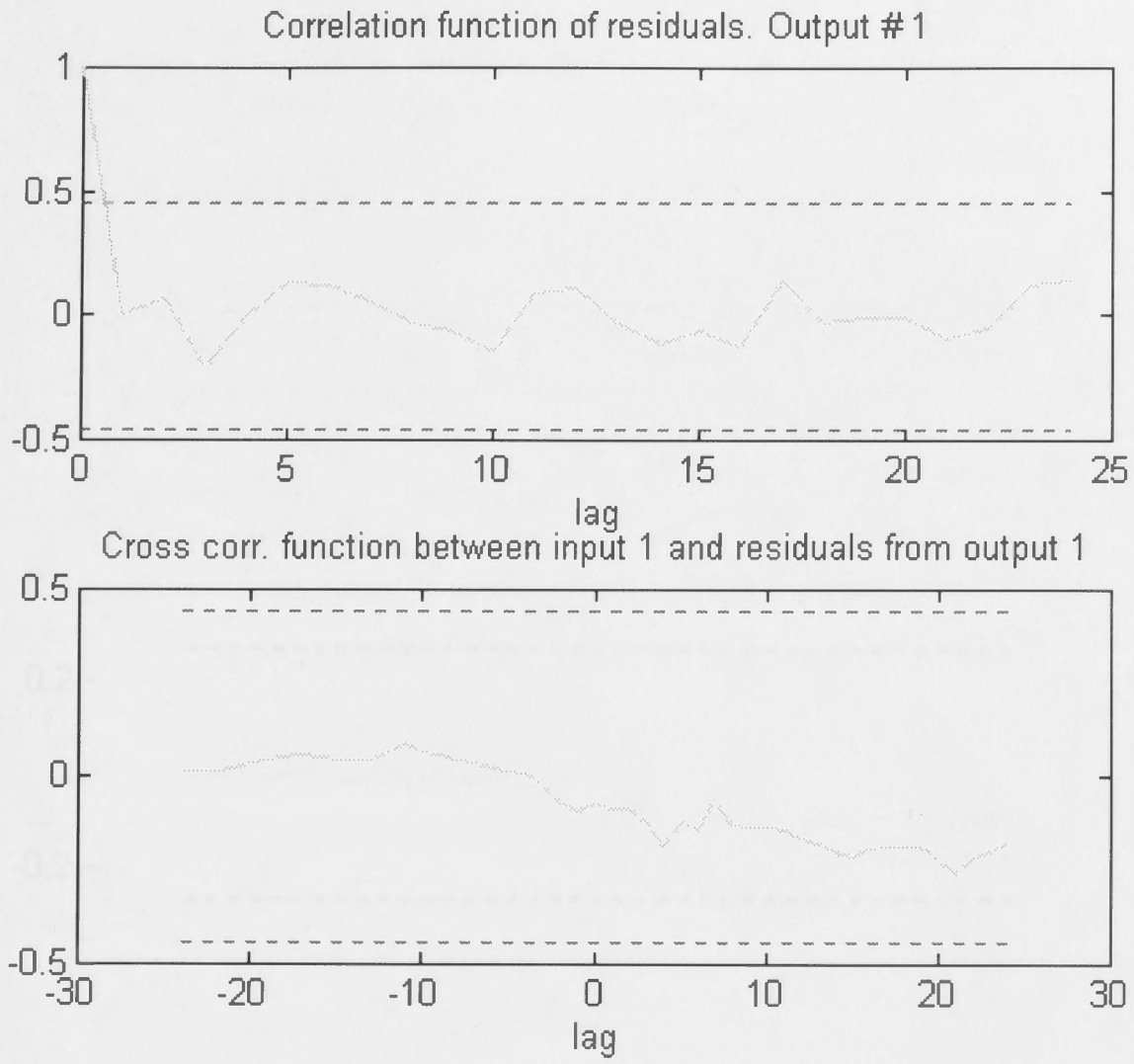


Figure 4A.2: Correlation functions for plot 2 replicate B

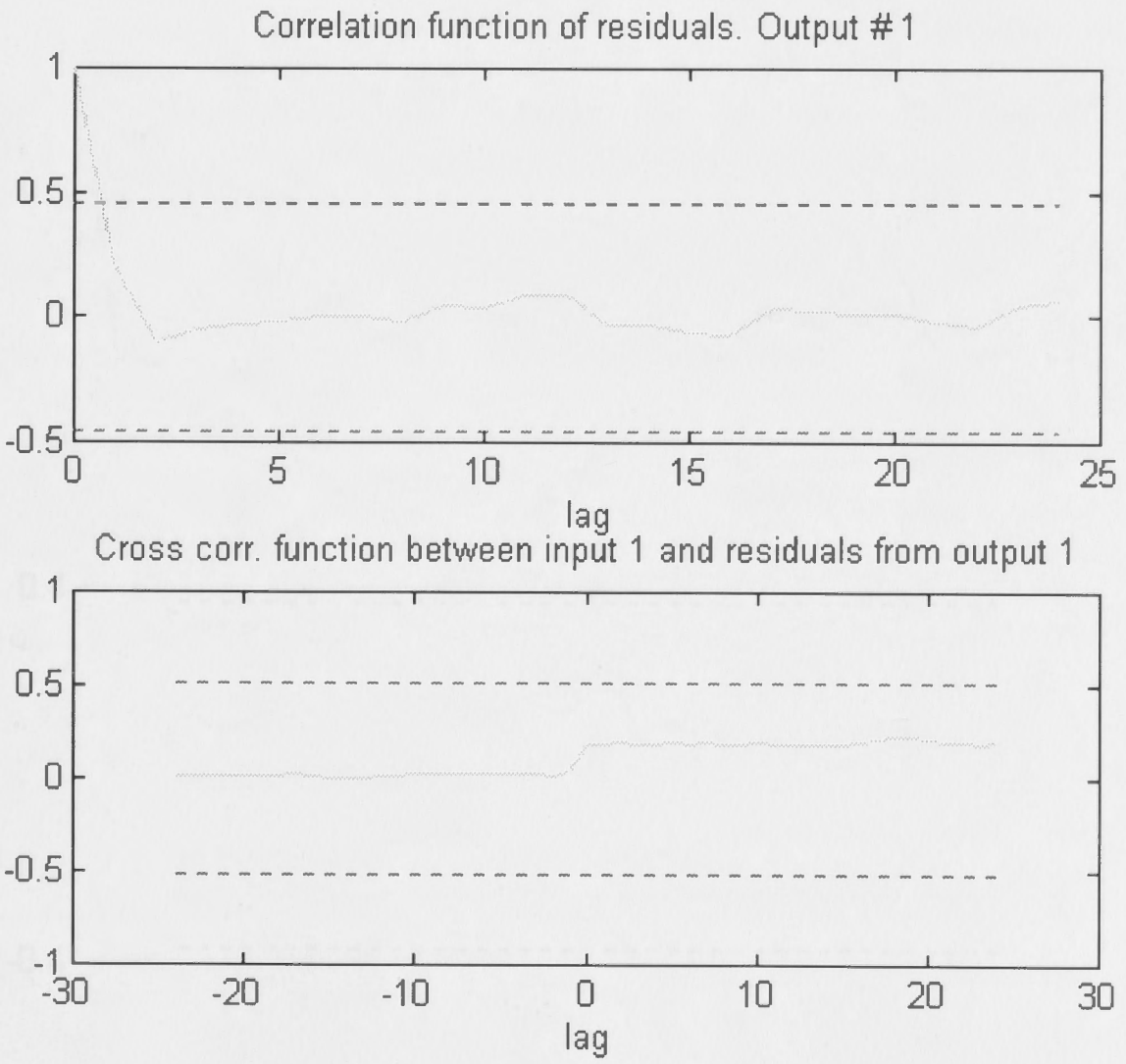


Figure 4A.3: Correlation functions for plot 3 replicate C

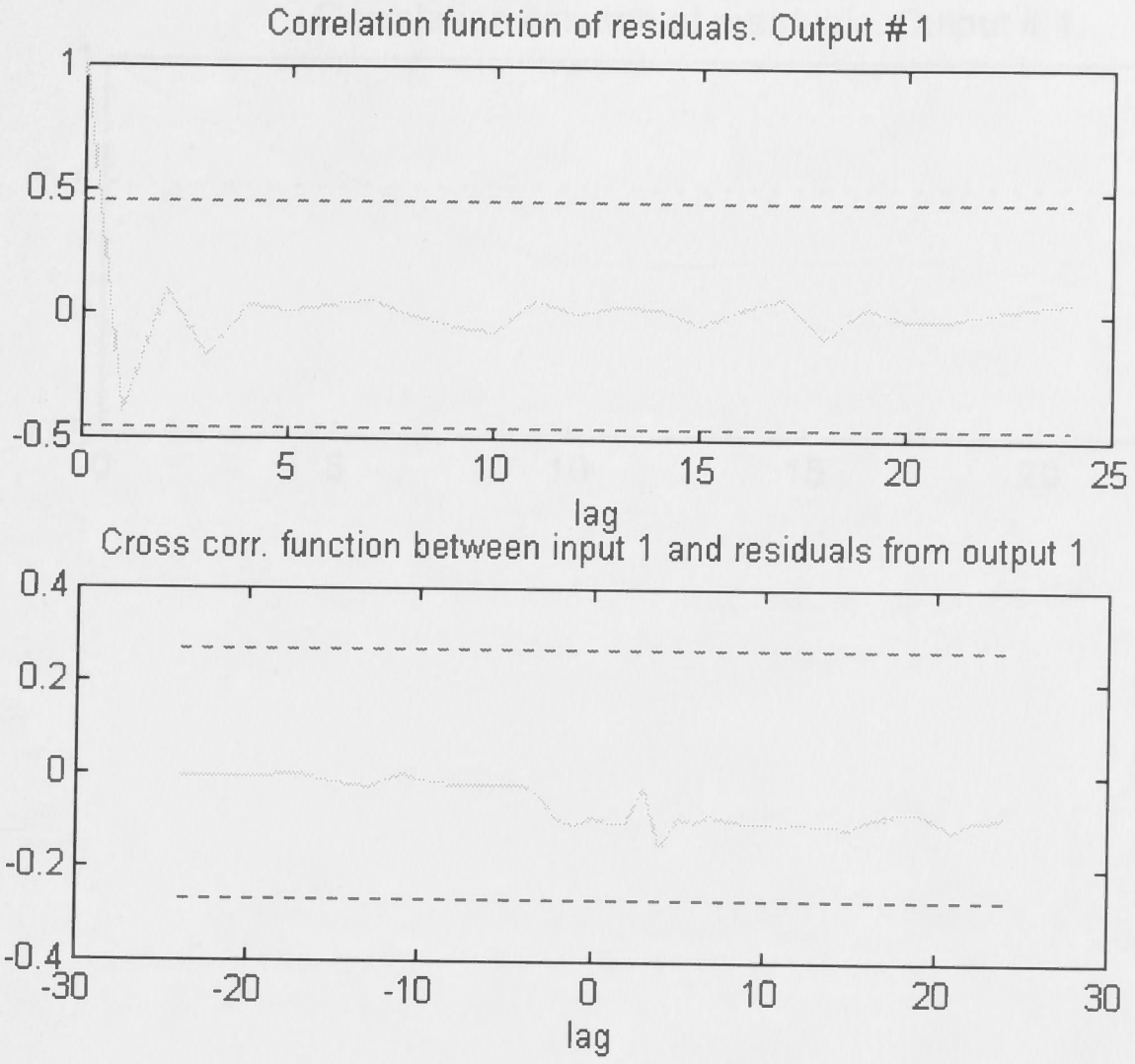


Figure 4A.4: Correlation functions for plot 4 replicate C

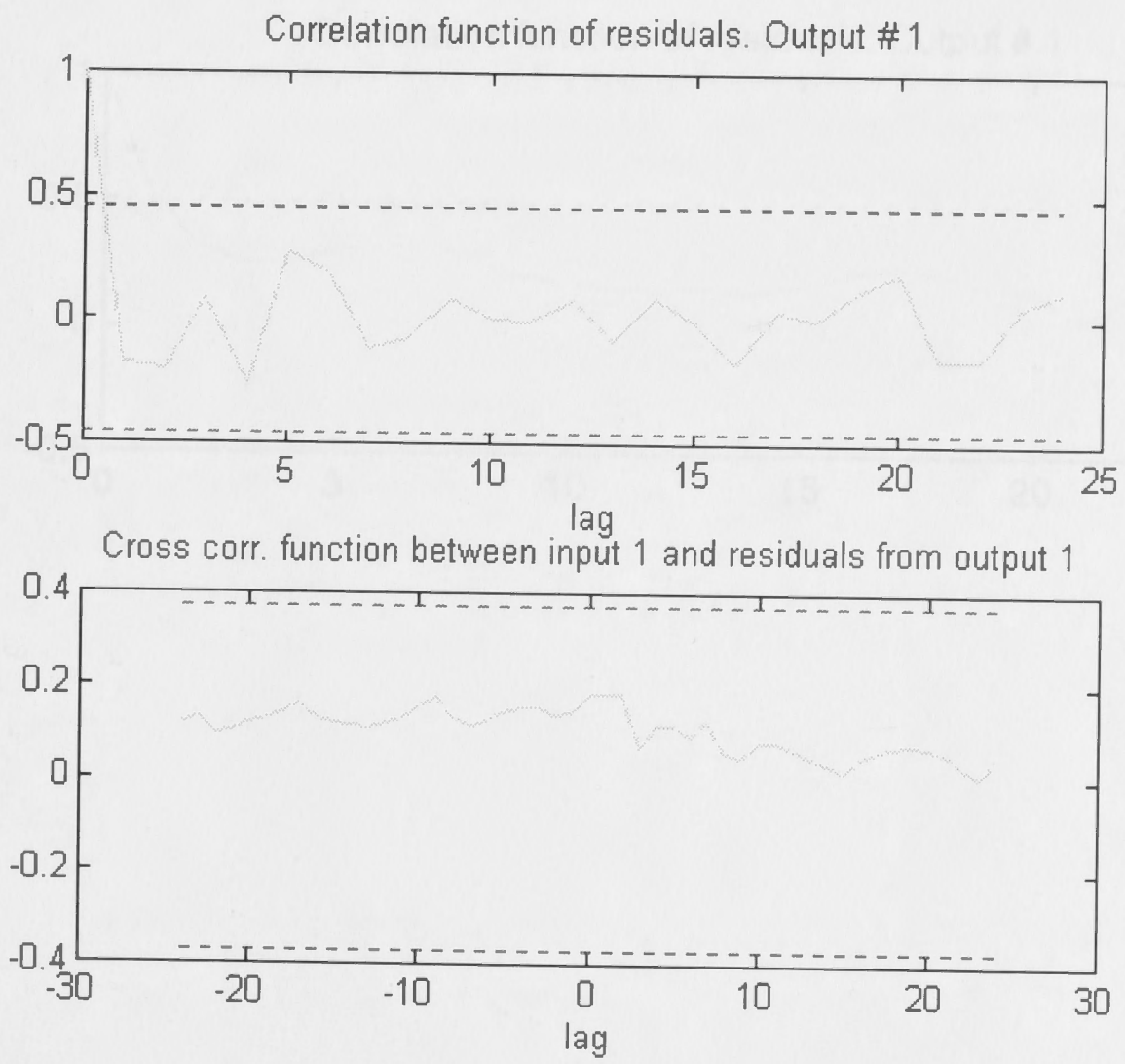


Figure 4A.5: Correlation functions for plot 5 replicate D

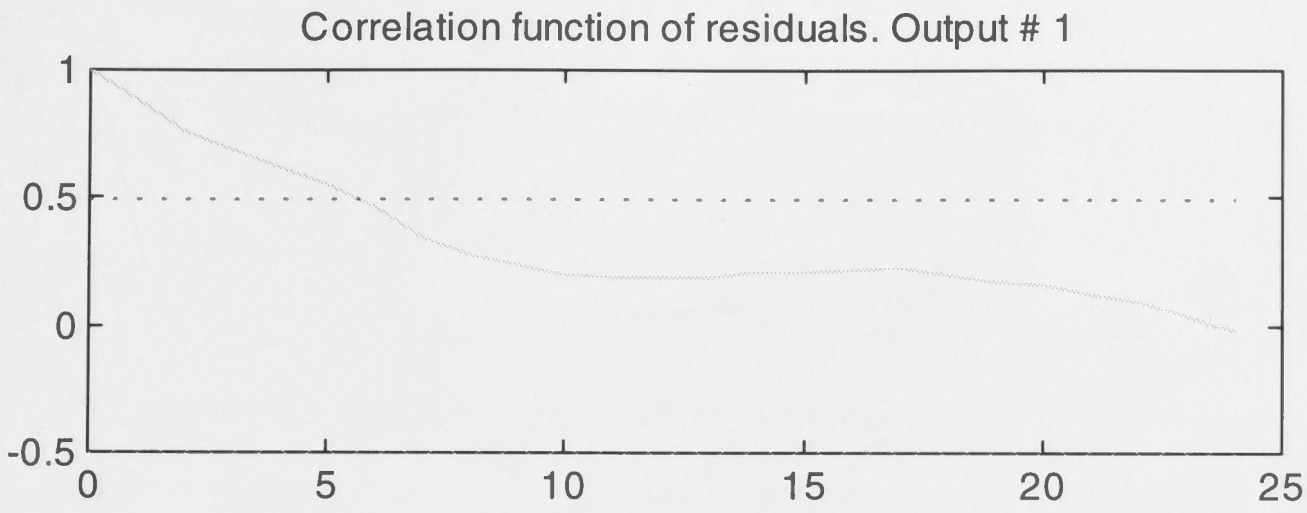


Figure 4A.6: Correlation functions for plot 6 replicate C

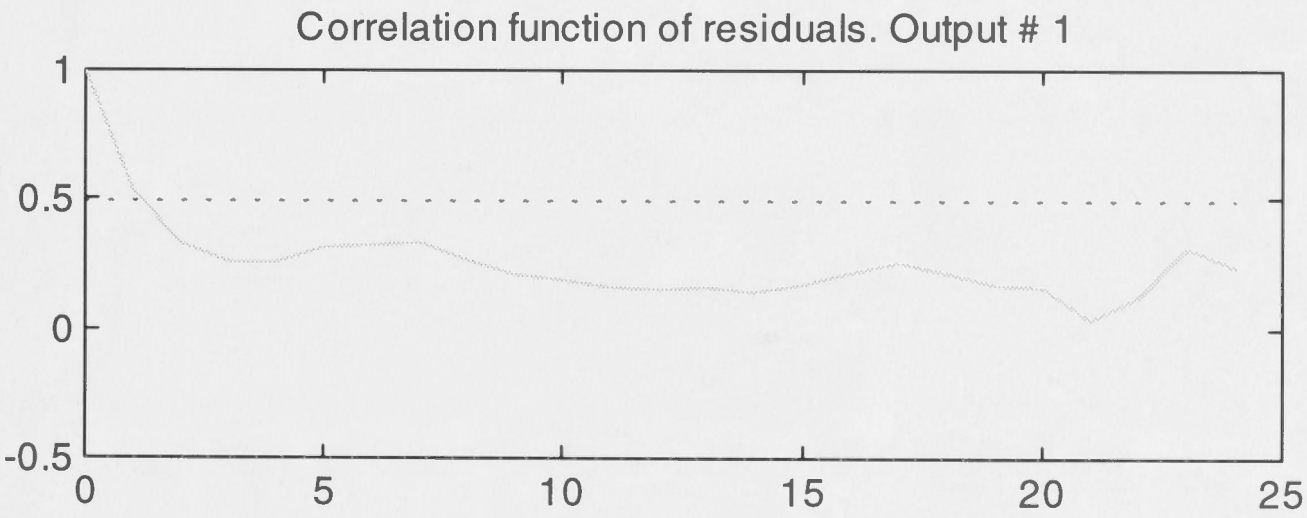


Figure 4A.7: Correlation functions for plot 7 replicate B

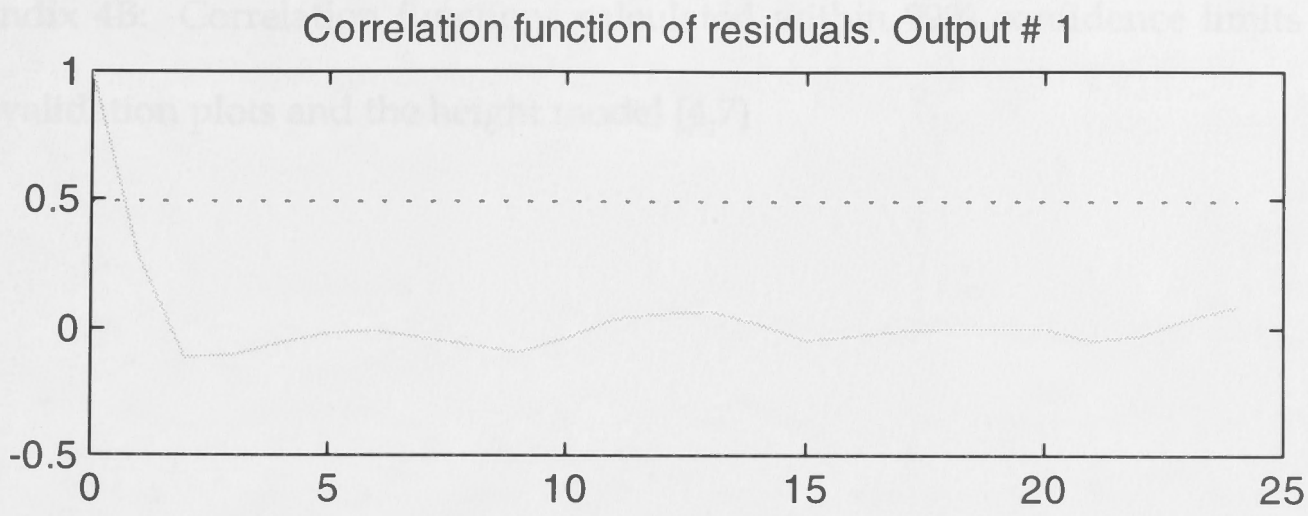


Figure 4A.8: Correlation functions for plot 8 replicate D

Appendix 4B: Correlation functions calculated within 99% confidence limits for the cross validation plots and the height model (4.7)

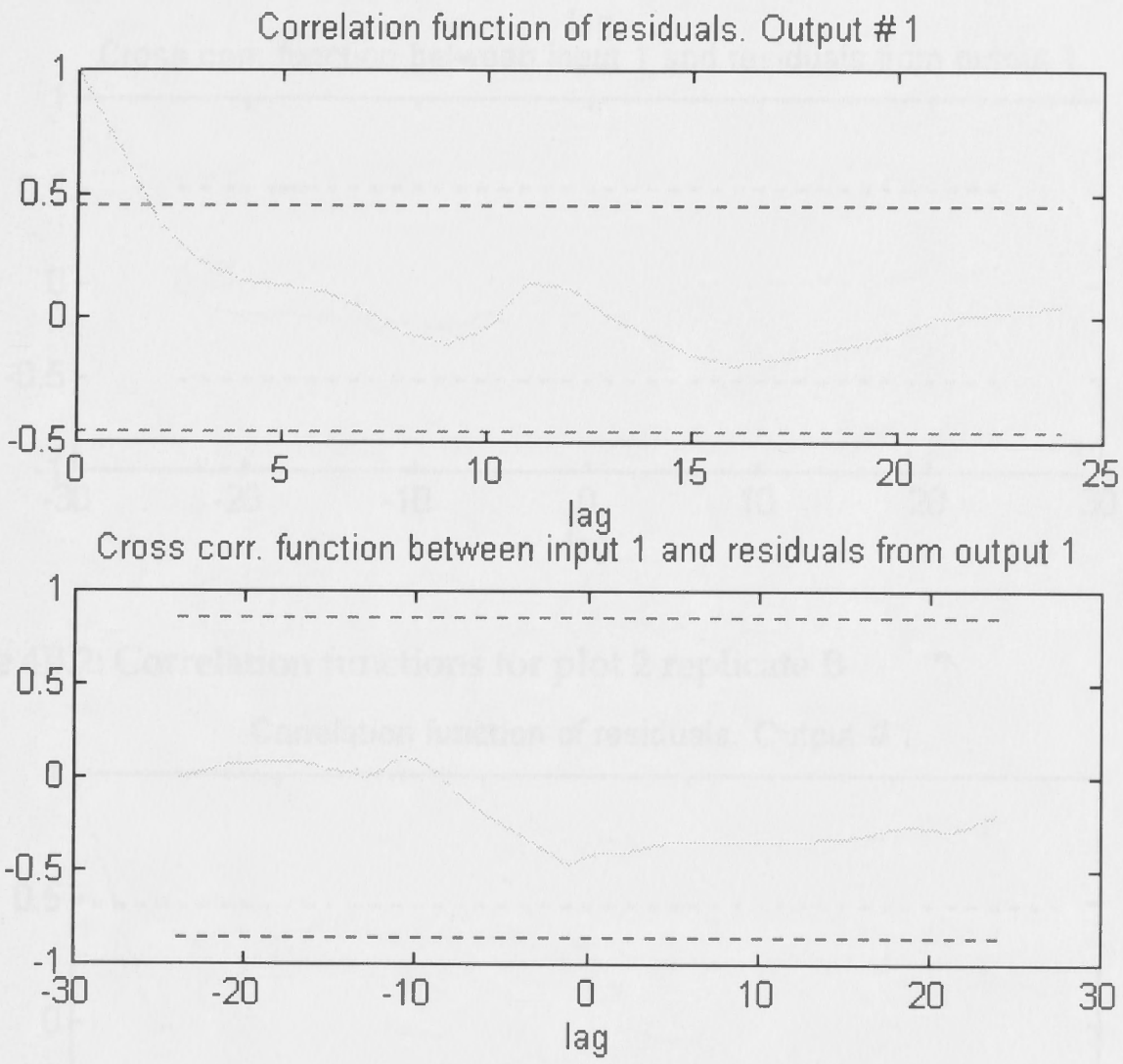


Figure 4B.1: Correlation functions for plot 1 replicate B

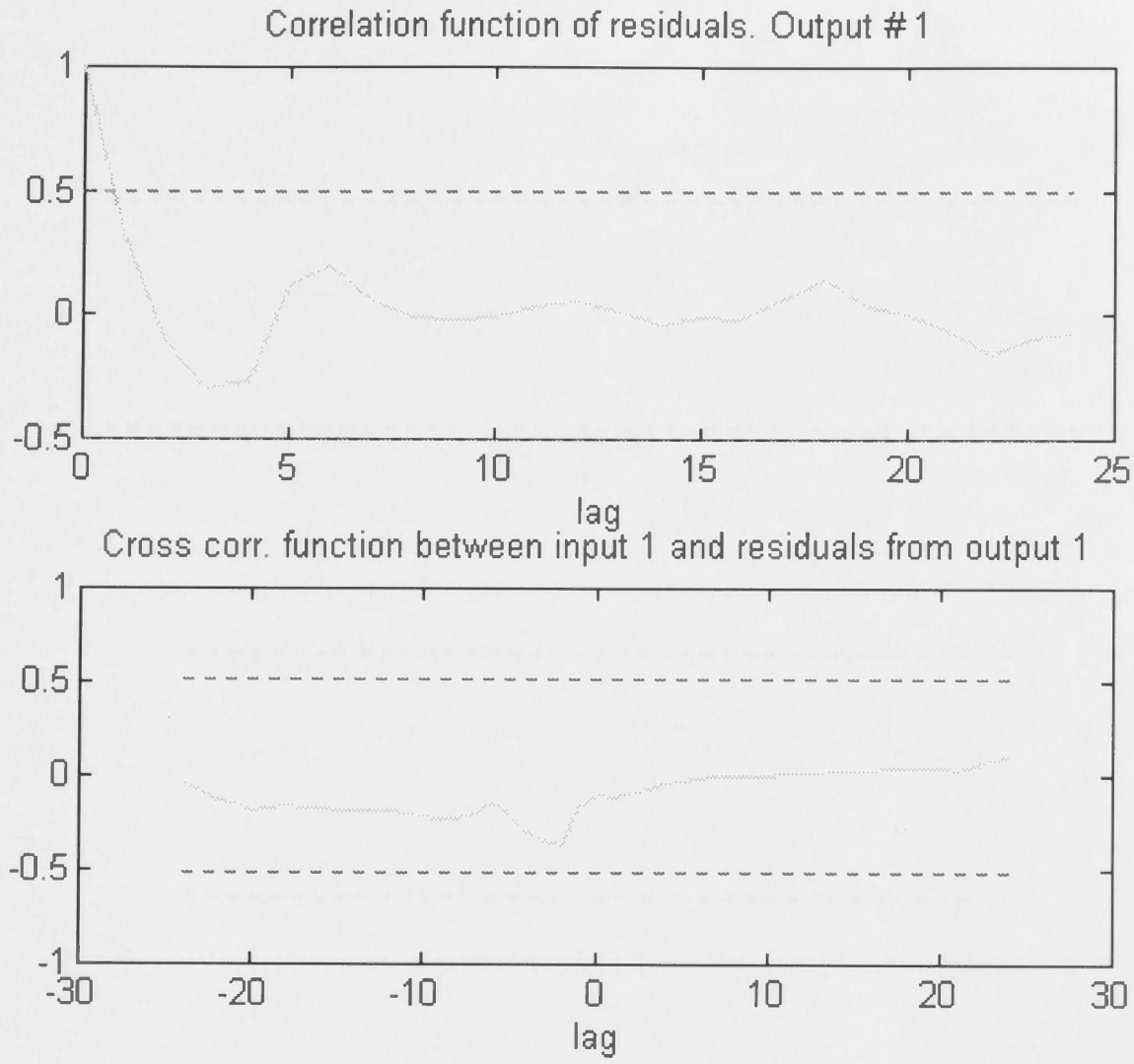


Figure 4B.2: Correlation functions for plot 2 replicate B

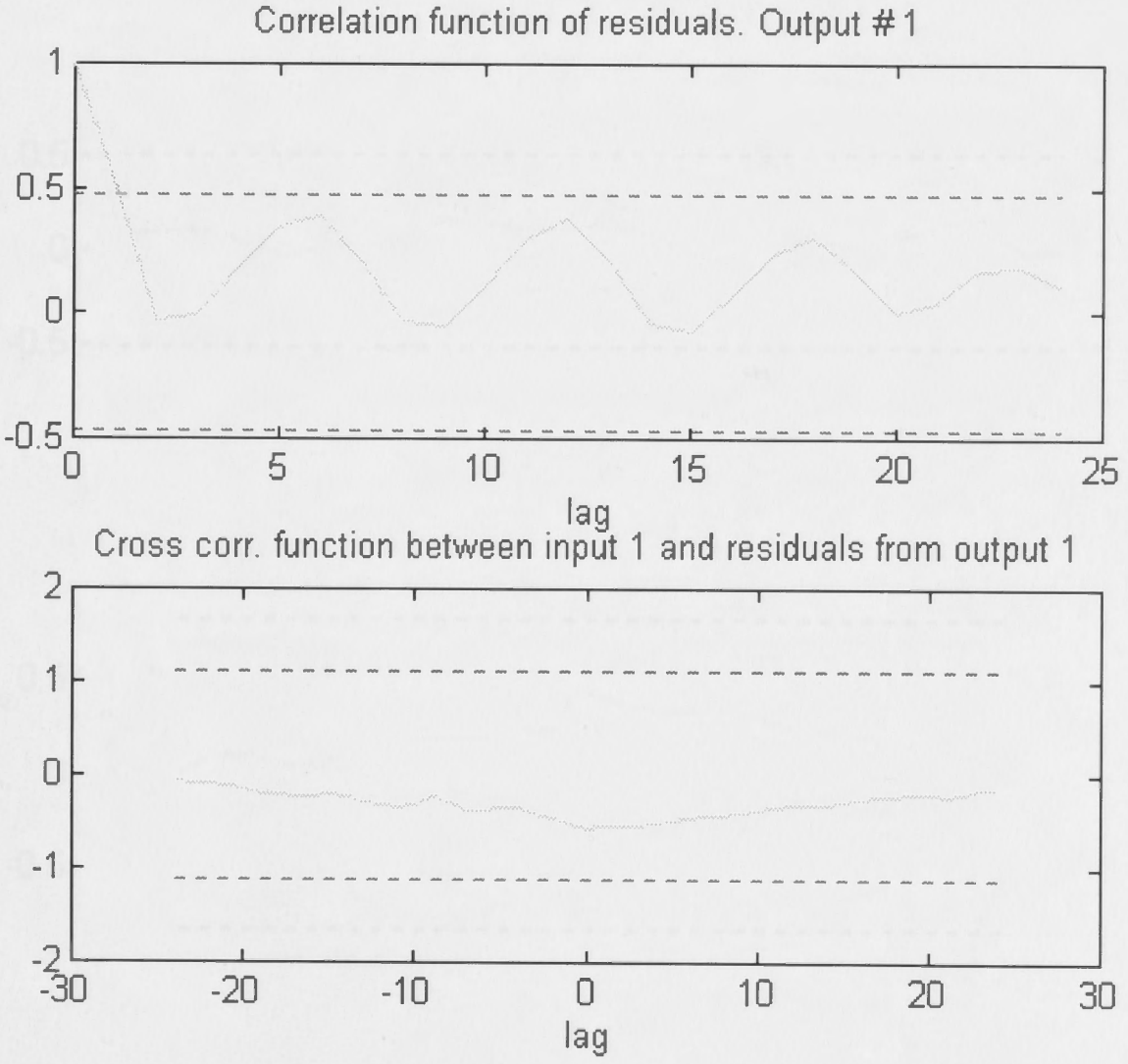


Figure 4B.3: Correlation functions for plot 3 replicate C

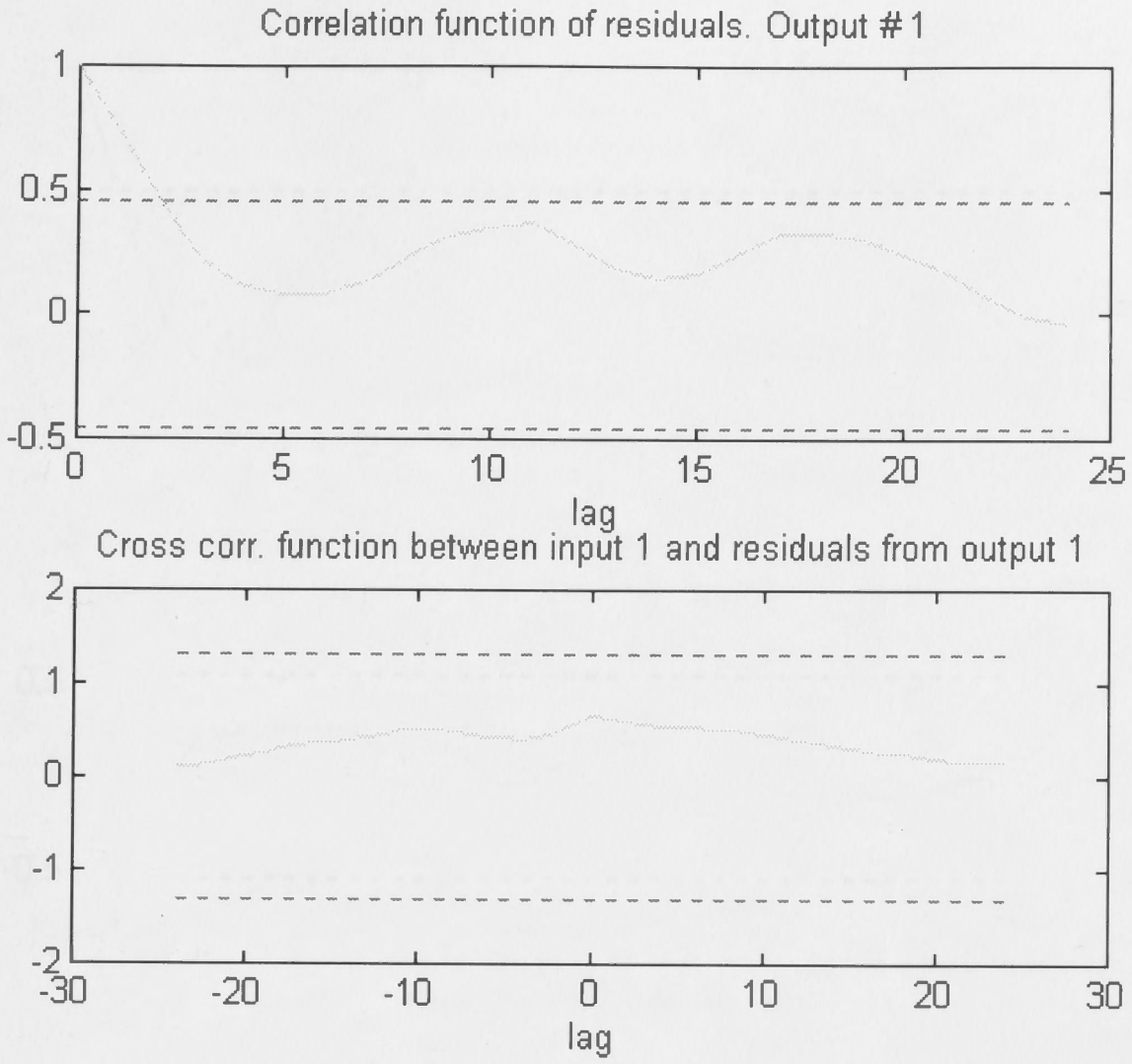


Figure 4B.4: Correlation functions for plot 4 replicate D

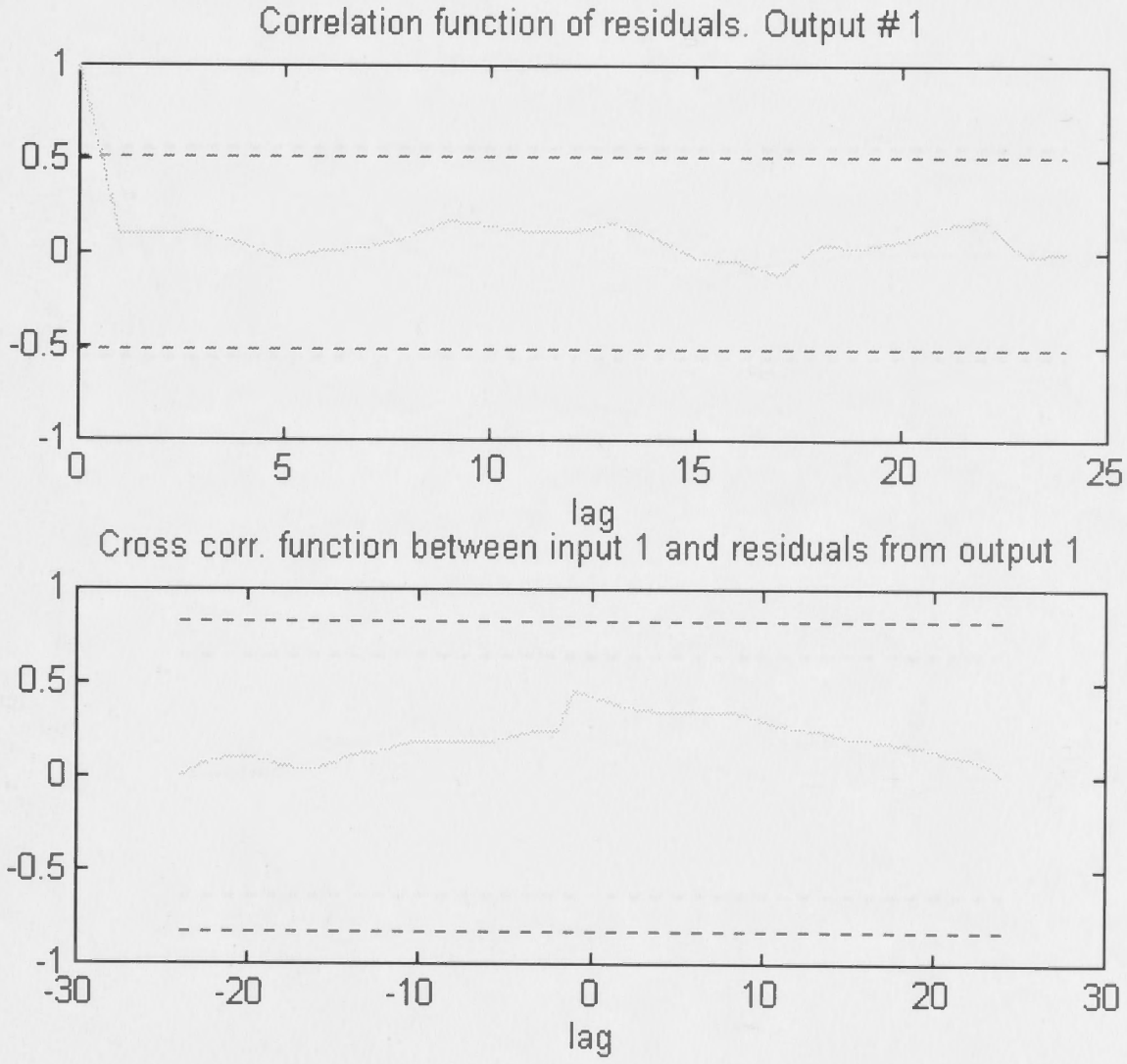


Figure 4B.5: Correlation functions for plot 5 replicate D

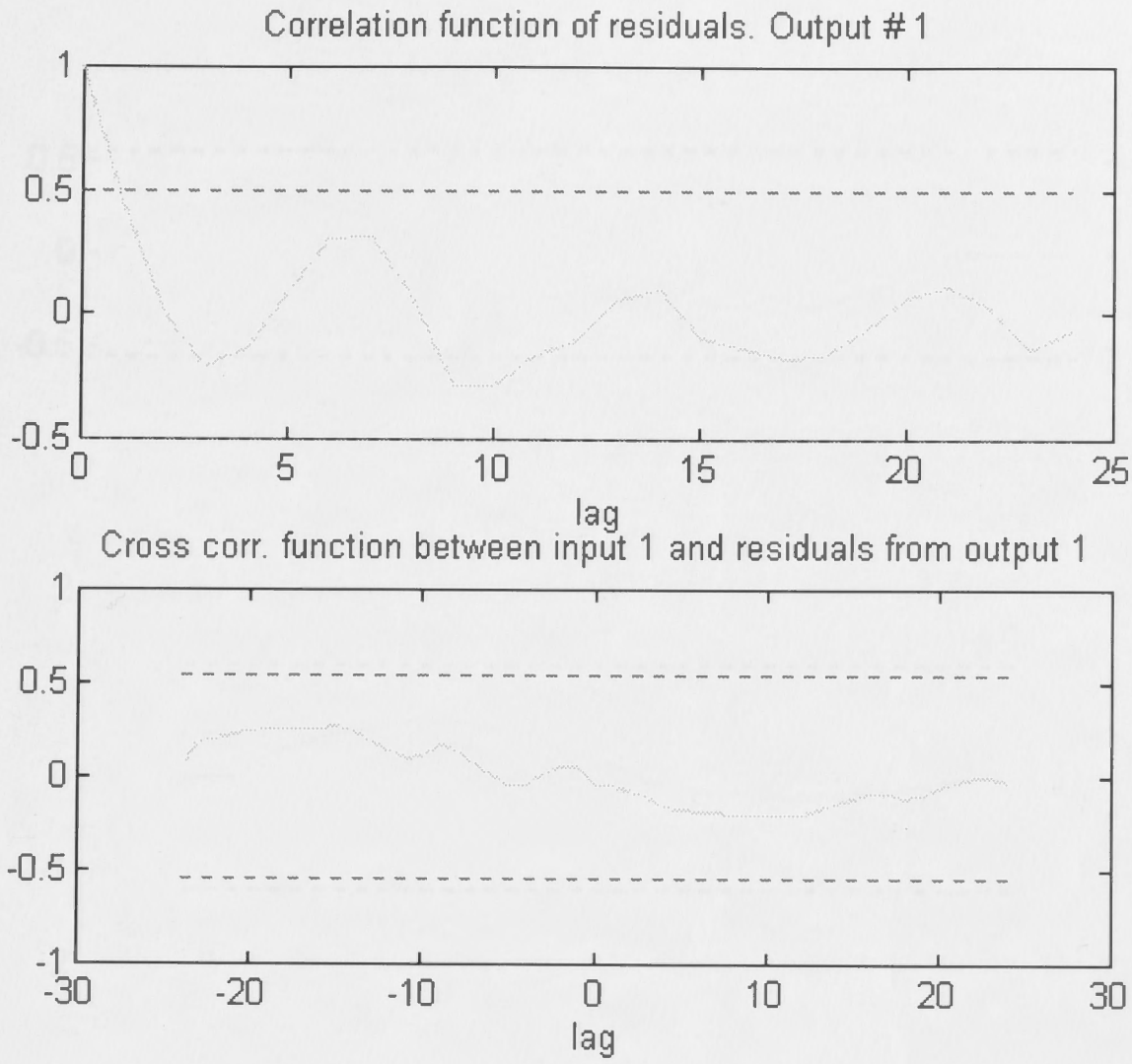


Figure 4B.6: Correlation functions for plot 6 replicate B

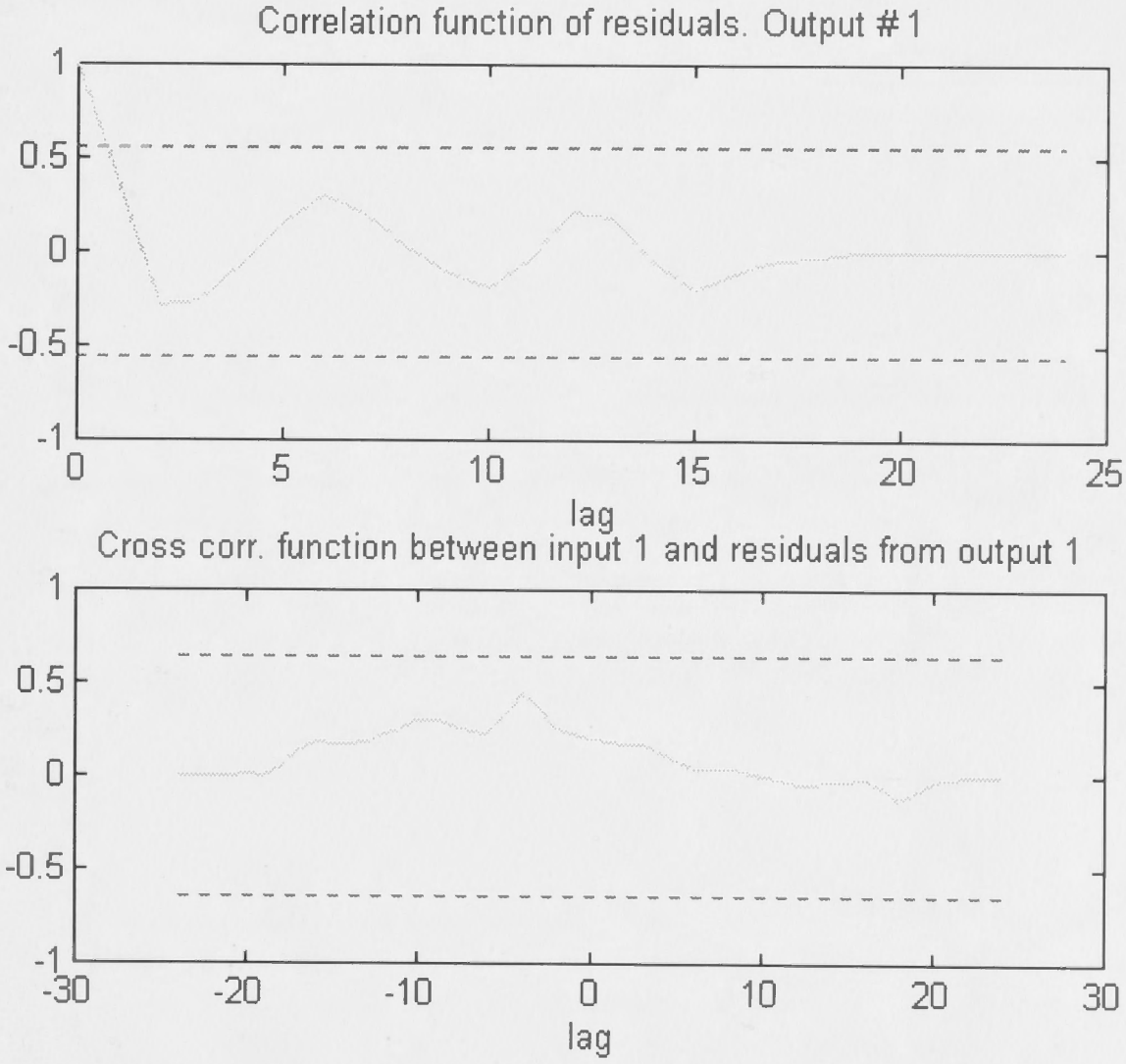


Figure 4B.7: Correlation functions for plot 7 replicate C

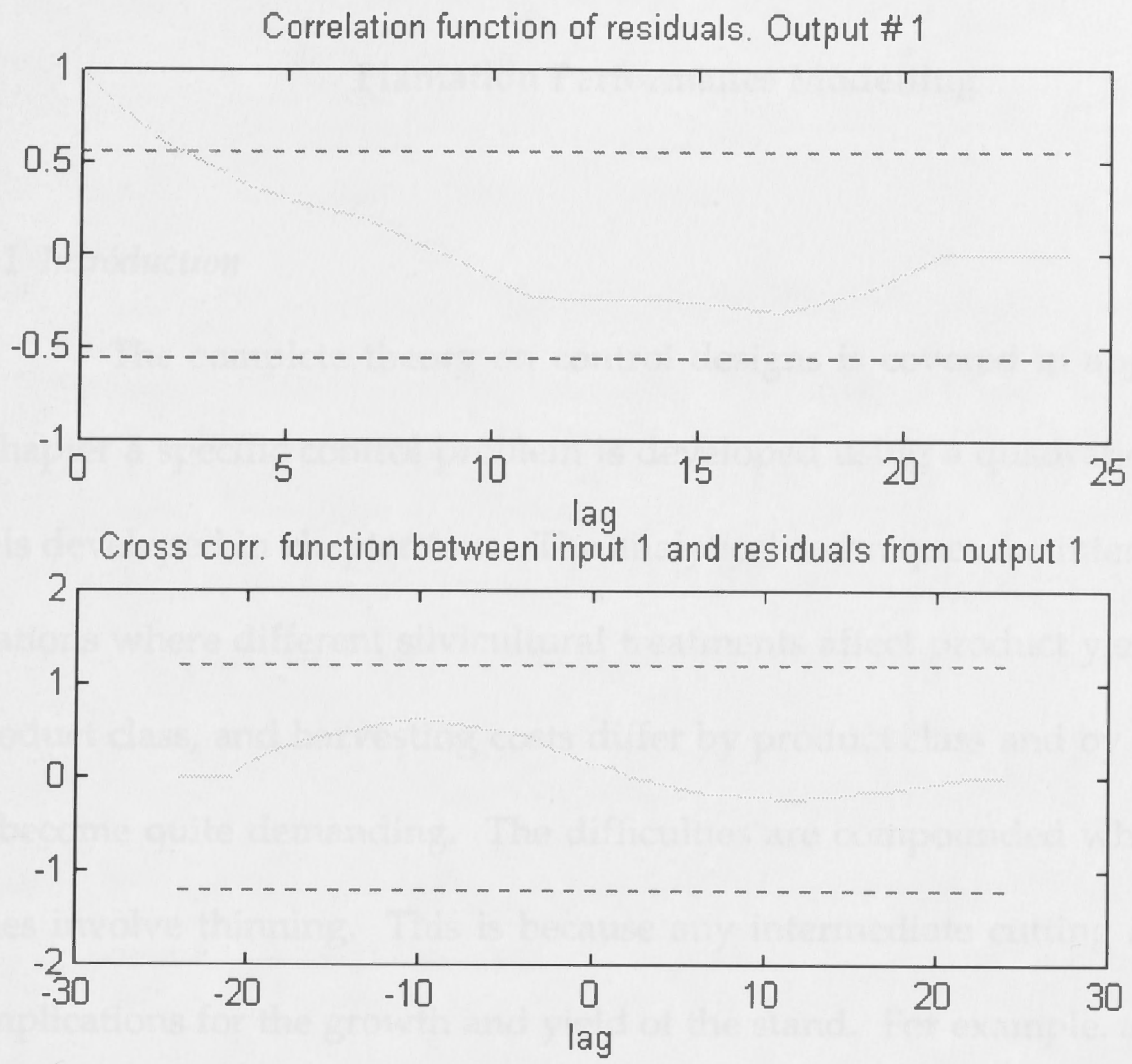


Figure 4B.8: Correlation functions for plot 8 replicate C

CHAPTER 5

Plantation Performance Modelling

5.1 Introduction

The complete theory on control designs is covered in appendix II and in this chapter a specific control problem is developed using a quadratic[1] functional and models developed in chapter four. The analytical techniques for intensively managed plantations where different silvicultural treatments affect product yields, prices differ by product class, and harvesting costs differ by product class and by volume per unit area, become quite demanding. The difficulties are compounded when management regimes involve thinning. This is because any intermediate cutting in a forest stand has implications for the growth and yield of the stand. For example, a heavy thinning in a young plantation significantly reduces the range of possible residual densities as the stand gets older. In general, each thinning decision in a stand affects all future growth, subsequent thinning decisions, and returns.

In other words forest managers face sequential or *interdependent* decision-making problems when planning the intermediate harvests in forest lands (Chen et al., 1980). This problem defines a multistage optimisation process which calls for a control design that may be solved by any one of the following solution techniques; calculus methods, experimental methods, linear and non-linear programming, calculus of variations, Lagrange multiplier methods, dynamic programming and the maximum principle (Fan and Wang, 1964).

---

[1] A quadratic form is a scalar function $V(x)$ of variables $x = [x_1, x_2, ..., x_n]$ defined by $x'Px$. The matrix P is of order n x n and is symmetric i.e. $P_{ij} = P_{ji}$ ($\forall\, i,j$, where $i \neq j$) so that $P' = P$ (Burghes and Graham, 1980).

Control or guidance usually refers to the directed influence on a dynamical system to achieve a desired performance (Luenberger, 1969). It is similar to planning approaches where an optimal procedure is determined for attaining a set of objectives. Such a system has an input/output description and the inputs $\{u\}$ are selected sequentially after observing past outputs $\{y\}$. The basic input/output description for a discrete linear time-invariant system with noise $\{e\}$ is given by equation (2.11). The state representation is utilised in the control design problem and it forms the relationship between the input, noise, and the output variables which are written as a system of first order (in this case) difference equations (see equations (3.16) and (5.4-6)) using an auxiliary state vector $\xi(t)$:

$$\xi(t+1) = F(\xi(t)) + Gu(t)$$

$$y(t) = \xi(t) \tag{5.1}$$

The optimisation criterion will take the form:

$$J(u) = \sum_{t=1}^{T} [\frac{u(t)}{x(t)} V(t)] \tag{5.2}$$

where

$x(t)$ = number of standing trees;

$u(t)$ = number of trees removed in thinning;

$V(t)$ = yield i.e. the total volume at time t;

$\frac{u(t)}{x(t)} V(t)$ = total volume harvested at time t;

What is required is to find a control sequence, $u(t)$, $t=1, 2, \ldots, T$, such that J is maximised.

5.2 Optimal Control

Yield is the volume function (4.11) expressed in m^3/ha:

$$V(t) = 0.4BA(t)H(t) \tag{5.3}$$

subject to the following system equations

$$x(t) = x(t\text{-}1) - u(t\text{-}1) \tag{5.4}$$

$$BA(t) = a_1(x(t\text{-}1))BA(t\text{-}1) + b_1(x(t\text{-}1)) \tag{5.5}$$

$$H(t) = a_2(x(t\text{-}1))H(t\text{-}1) + b_2(x(t\text{-}1))f(BA(t), x(t)) \tag{5.6}$$

where the initial conditions, $x(0)$, $BA(0)$ and $H(0)$ are given.

$x(t)$, $BA(t)$ and $H(t)$ form the state variable $\xi(t)$ where $x(t)$ is the number of trees per hectare, $BA(t)$ the basal area (see model (4.1)) and $H(t)$ the height function (see model (4.7)). Figure 5.1 shows the diagrammatic representation of the input/state/output structure for the control model (5.3-6). The parameters a_1, a_2, b_1 and b_2 are dependent on x:

$$a_1(x) = 0.93 + 0.01\frac{x}{1000} - 0.047\left(\frac{x}{1000}\right)^2 + 0.01\left(\frac{x}{1000}\right)^3 \tag{5.7}$$

$$b_1(x) = 2.32 + 5.24\frac{x}{1000} - 0.35\left(\frac{x}{1000}\right)^2 \tag{5.8}$$

$$a_2(x) = \begin{cases} 0.782 & x \geq 1000 \\ 0.85 & 1000 > x \geq 400 \\ 0.913 & 400 > x \geq 100 \end{cases} \tag{5.9}$$

$$b_2(x) = \begin{cases} 0.19 + 0.03\dfrac{x}{1000} & x \geq 1000 \\ 0.095 + 0.05\dfrac{x}{1000} & 1000 > x \geq 400 \\ 0.035 + 0.1\dfrac{x}{1000} & 400 > x \geq 100 \end{cases} \tag{5.10}$$

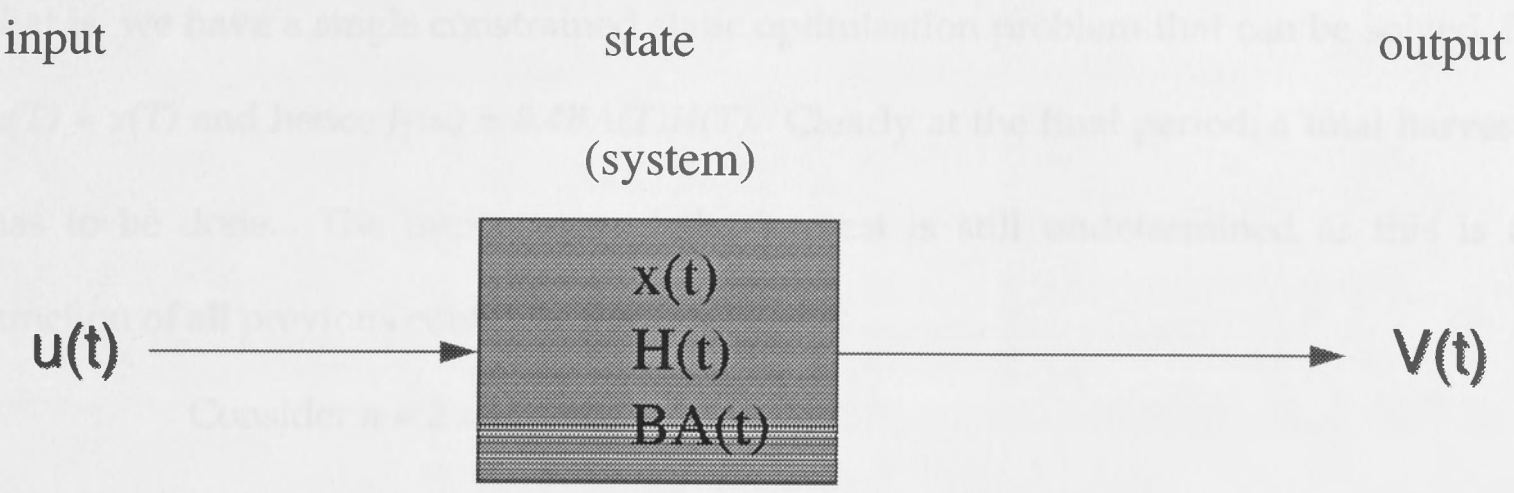


Figure 5.1: State space representation of the control model (5.3-6).

$f(BA(t), x(t))$ in equation (5.6) is the quadratic mean diameter, derived from (5.5) and hence is expressed as follows:

$$f(BA, x) = c\sqrt{\frac{BA}{x}} \tag{5.11}$$

where $c = 1/\sqrt{(pi/4)/10\,000}$. (5.12)

$u(t)$ in equation (5.4) is the number of trees thinned in year t.

Let $J_n(u)$ be defined as the maximum achievable total volume when only n periods remain. Thus

$$J_n(u) = \max_{u(t)} \sum_{t=T-(n-1)}^{T} \frac{u(t)}{x(t)}[0.4BA(t)H(t)] \tag{5.13}$$

subject to the same constraints of (5.3) for $t \geq T\text{-}(n\text{-}1)$ with upper and lower bounds on the control, $u_{min} \leq u(t) \leq u_{max}, \forall t \in [t_0, T]$. Silvicultural knowledge is required to determine the bounds on the control. The control constraints can be varied for each time, t, depending on management requirements.

For $n = 1$,

$$J_1(u) = \max_{u(T)} \left[\frac{u(T)}{x(T)} 0.4BA(T)\, H(T)\right] \tag{5.14}$$

that is, we have a single constrained static optimisation problem that can be solved, if $u(T) = x(T)$ and hence $J_1(u) = 0.4BA(T)H(T)$. Clearly at the final period, a total harvest has to be done. The importance of the harvest is still undetermined as this is a function of all previous control actions.

Consider $n = 2$ and

$$J_2(u) = \max_{u(T-1)} [\frac{u(T-1)}{x(T-1)} 0.4BA(T\text{-}1)H(T\text{-}1) \ ...$$

$$+ 0.4BA(T)H(T) \ ]$$

$$= \max_{u(T-1)} [\frac{u(T-1)}{x(T-1)} 0.4BA(T\text{-}1)H(T\text{-}1) \ ...$$

$$+ 0.4[a_1(x(T\text{-}1))BA(T\text{-}1) + b_1(x(T\text{-}1))]...$$

$$[a_2(x(T\text{-}1))H(T\text{-}1) + b_2(x(T\text{-}1))f(BA(T\text{-}1), x(T\text{-}1))]] \qquad (5.15)$$

and so on, until $n = T+1$ and the original problem has been solved. This is the method of dynamic programming, sometimes called *backwards induction*. $J_2(u)$ becomes a function of $u(T\text{-}1)$ and variables that cannot be affected by $u(T\text{-}1)$, namely $x(T\text{-}1)$, $BA(T\text{-}1)$ and $H(T\text{-}1)$; $u(T\text{-}1)$ can only affect the future state values. Therefore, $J_2(u)$ is optimised with respect to $u(T\text{-}1)$. This optimal solution is by necessity a function of $\xi(T\text{-}1)$. It follows that the optimal control policy found thus is a function of the state variables, i.e.

$$u(T) = x(T)$$

$$u(T\text{-}1) = u^*(T\text{-}1, \xi(T\text{-}1))$$

. .

. .

$$u(2) = u^*(2, \xi(2))$$

$$u(1) = u^*(1, \xi(1))$$

The actual computation is cumbersome especially when constraints are specified. However, the above problem was solved using a software package called DMISER3 (Jennings et al., 1990). This program is generally used for solving discrete-time optimal control and optimal parameter selection problems. DMISER3 employs the solution technique of maximum principle (Pontryagin, 1959 a;b;c). Note that maximum principle and dynamic programming are essentially the same (Fan and Wang, 1964). The user need only apply a set of explicit FORTRAN expressions for the functions and their derivatives with respect to the state, control and parameters. The values of the parameters and variables that define the problem and its accuracies are coded into a separate data file (see appendix III).

The control model can be set up as a constrained optimisation problem. A MatLab program that uses constrained optimisation for the control problem is found in appendix III.

5.2.1 Optimum Management Strategies

Plantation forestry can have relatively simple objectives, such as producing wood in a few classes from monoculture stands established and managed for this specific purpose. In order for managers to take into account other objectives such as recreation, water catchment and conservation values and to respond promptly to sudden changes in world trade trends, technology and availability of labour, it is vital to have a reliable optimisation base that will help to foresee consequences of management options. In the following subsection silvicultural regimes for volume production and value production have been suggested by using DMISER3. The regimes are based primarily on plantations of *P. patula* in South Africa, of very good productivity, that correspond to site quality I (Van Laar, 1976).

Maximum production regimes

Maximum production refers to the maximum wood products which a species can yield on a site. Therefore, the primary objective is to maximise volume production where the financial gains take second priority. In some situations volume production can be equated to maximum financial return, where there is no differentiation in the stumpage or royalty rates with respect to size or quality of product. DMISER3 was used to determine a silvicultural regime to meet the objective of volume production. The exercise involved two stages:

(1) optimisation without thinning constraints; and

(2) optimisation with thinning constraints.

The initial planting density bounds were set at 800-2000 stems/ha, 0-1000 stems/ha for the thinning intensity (control) and the final crop number was not constrained. A typical DMISER3 output is shown in appendix 5A for a 25 year rotation of a volume production regime. Since each year interval was defined as a stage, DMISER3 optimised by thinning less than ten trees in some years. Successive runs were done at each one year increment in rotation length from 25 years. This was done to determine the optimal rotation length. There was no change in the regime until after the age 35 when the thinning intensity and timing radically changed with more trees retained as a final crop. The volume mean annual increment was calculated to be an average of 6 m3/ha/yr between the ages 25-35 years. Thus a delayed thinning after the age of 25 would not contribute substantially to volume increment. This minimal response to volume increment at late age is what is observed in a real situation and the control model reflects the same characteristics. Shepherd (1986) observed the same responses and recommended a late age thinning and fertilisation to boost the volume increment.

The volume production regime obtained is summarised below:

Age (yrs)	Remaining stems/ha
0	2000
6-9	1268
11-14	1060
19	979
25-35	338

In the second stage of the exercise, three thinning times were specified based on when the major thinnings occurred in the first stage. The following regime was obtained:

Age(yrs)	Remaining stems/ha
0	2000
7	1275
12	1034
19	396
25-35	0

It is very difficult to comment on the above regime unless some knowledge on the available markets, company objectives and silvicultural considerations such as wind stability is available. However, this is the regime that produces the maximum total volume over a rotation of 25 years that can be delayed to 35 years with minimal volume change.

Maximum value regime

Maximum value regimes rely on there being a price-size gradient whereby larger logs command the highest prices (Shepherd, 1986) and preferably an additional revenue component for quality is in place. A good example is where large pruned logs are sold as veneer logs at a premium price. Therefore, the objective is to obtain the maximum possible volume of these quality products in the shortest possible time,

without concern for any loss in total productivity that this might entail. The objective function in DMISER3 was then defined as follows:

$$J_N = \sum_{t=1}^{T} \frac{u(t)}{x(t)} V(t) \frac{BA(t)}{x(t)} \tag{5.16}$$

where $\frac{BA(t)}{x(t)}$ = average basal area per tree.

Average basal area per tree is a common basis on which standing timber is valued and this simple size characteristic can be used in place of a frequency distribution. The total monetary value calculated via the individual tree distribution will usually differ in a consistent way from that calculated via the mean tree size (Johnston, et al., 1967), depending on the type of thinning. Therefore, the mean tree size is, in practice, almost as efficient as a frequency distribution of tree size, in calculating the total relative value of a thinning or felling at a point in time. However, the influence of a thinning type on a frequency distribution has not been considered in this thesis.

An initial run with constraints on control of 0-1000 stems/ha, 800-2000 stems/ha for the initial planting density and a constraint on the final crop of 150-1000 stems/ha, generated a regime that would cause a major concern for wind stability of the residual trees after thinning. This led to a constraining of the final crop density to 250-1000 stems/ha. The following regime was obtained with only the major thinnings shown:

Age(yrs)	Remaining stems/ha
0	2000
7	1555
9-14	1073
22	751
23	434
25-35	250

In the second stage three thinnings were specified and the control was constrained further towards the end of the regime to minimise heavy thinning:

Age(yrs)	Remaining stems/ha
0	2000
7	1275
13	450
21	250
25-35	0

It is clear that for volume production, more trees have to be thinned at a younger age of the forest stand and relatively less have to be retained for a final crop. For value production, heavy thinnings are delayed until the trees have acquired a fairly large average diameter and hence wind stability has to be considered at all times. Silvicultural concerns can be raised and incorporated in the control model fairly quickly. DMISER3 will automatically produce an output that will satisfy these silvicultural concerns.

5.3 Sensitivity Analysis

It is important that a control system be insensitive or robust with respect to measurement errors, plant disturbances and modelling errors. The general topic of system characteristics changing with system parameter variations is called sensitivity. In order to measure sensitivity adequately, a fictitious thinning strategy with an initial stand density of 1100 stems/ha was derived from DMISER3 for a volume production control problem.

The a_1, a_2, b_1 and b_2 parameters from the control model (5.3-6) were then individually and independently varied at plus or minus their standard deviations obtained from system identification, and thinning strategies derived for the same

volume production control problem. The standard deviations in a_2 and b_2 were a result of the smoothed equations (see Appendix III). Figure 5.2 shows the different residual densities that were derived:

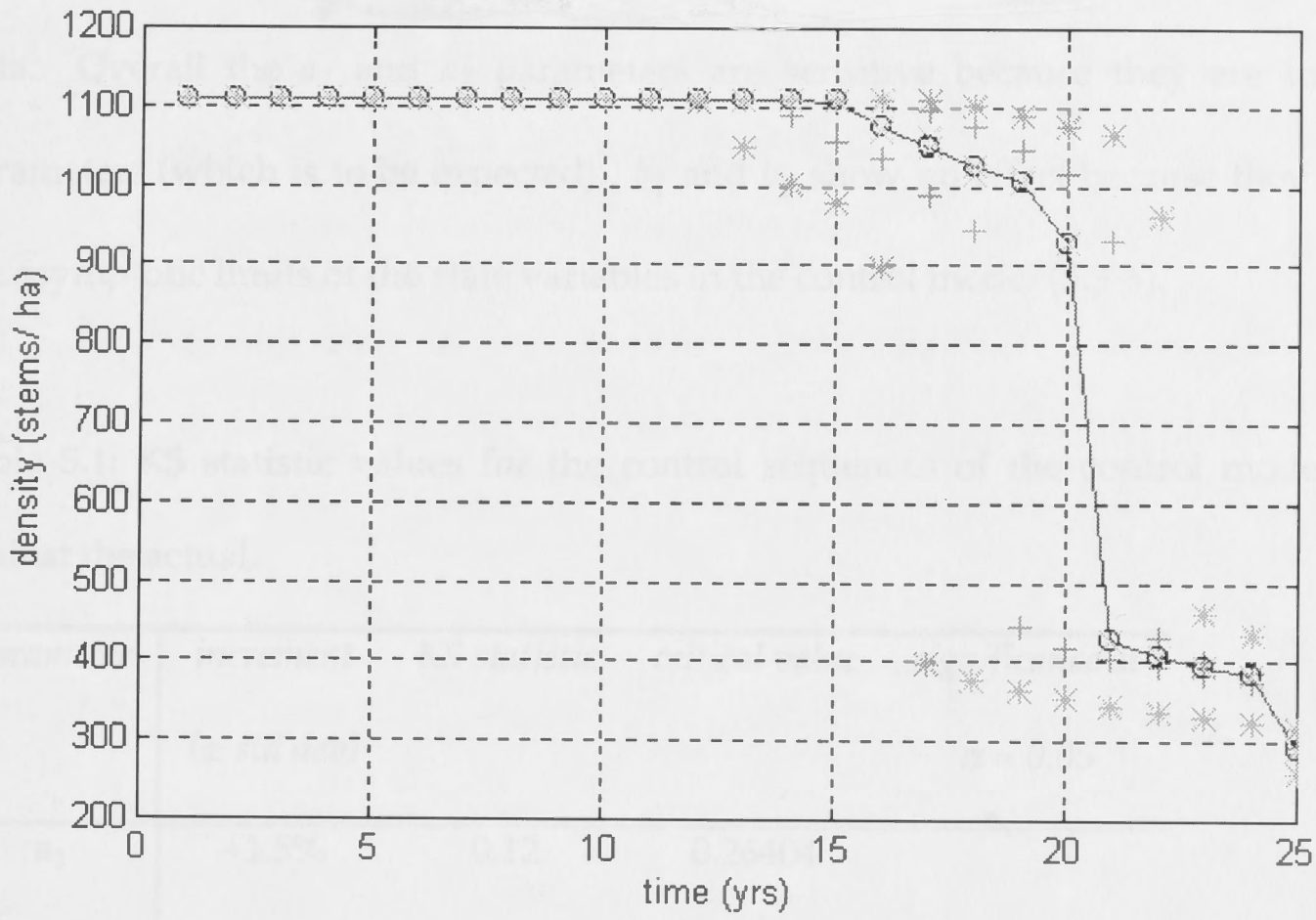


Figure 5.2: Residual densities for the volume production problem where,

_____ = actual residual density

++++++ = residual density with a_1 at $\pm 1.5\%$ standard deviation

oooooo = residual density with b_1 at $\pm 7.8\%$ standard deviation (no effect)

******** = residual density with a_2 at $\pm 5.6\%$ standard deviation

xxxxxx = residual density with b_2 at $\pm 5\%$ standard deviation (no effect)

The Kolmogorov Smirnov (KS) test was used to measure the overall difference between the actual strategy and each of the strategies derived from varying the control model (5.3-6) parameters, and the results are shown in Table 5.1. The KS statistic was used to measure the maximum value of the absolute difference between two thinning strategies (Press et al., 1992). If the calculated KS statistic is greater than

the critical value the functions are considered significantly different and this only occurred with the a_2 parameter of the height function. The sensitivity of the a_2 parameter can be attributed to the fact that the height function was piecewise continuous and the points of discontinuity were coarse because of the nature of the data. Overall the a_1 and a_2 parameters are sensitive because they are the shape parameters (which is to be expected). b_1 and b_2 show no effect because they indicate the asymptotic limits of the state variables in the control model (5.3-6).

Table 5.1: KS statistic values for the control sequences of the control model (5.3-6) against the actual.

parameter	*increment (± std dev)*	*KS statistic*	*critical value*	*significance at $\alpha = 0.05$*
a_1	+1.5%	0.12	0.26404	
	-1.5%	0.08	0.26404	
b_1	+7.8%	0.04	0.26404	
	-7.8%	0.04	0.26404	
a_2	+5.6%	0.28	0.26404	*
	-5.6%	0.2	0.26404	
b_2	+5%	0.04	0.26404	
	-5%	0.04	0.26404	

5.4 Comparison of DMISER3 regimes and current South African P. patula regimes

Van Laar (1976) discussed some of the earlier regimes (from 1940) for pine sawtimber in South Africa and the recommended remaining stems/ha were as follows:

Site quality 1		*Site quality 2*		*Site quality 3*	
Age (yrs)	stems/ha	Age (yrs)	stems/ha	Age (yrs)	stems/ha
0	1310	0	1310	0	1310
10	820	6	520	6	740
15	540	18	370	14	370
20	370	23	300	20	250
25	300	40	0	50	0
30	0				

From the early 1970's the Department of Forestry (South Africa) introduced a standard regime for pine sawtimber across all sites, in order to reduce the oversupply in pulpwood and to make the first thinning profitable (Wessels, 1977). This regime was:

All sites

Age (yrs)	Remaining stems/ha
0	1372
8	650
13	400
18	250
25-40	0

The private sector also initiated some revised thinning regimes for pine sawtimber, eg. the 'banana' regime (Bredenkamp, Venter and Haigh, 1983). There was

an oversupply of residual roundwood, and they wanted to reduce the sawtimber rotation. The recommended regime was:

All sites

Age (yrs)	Remaining stems/ha	
0	1111	
4-5	650	(Along with pruning to 1.5m when dominant height = 3.5m)
13	400	
25	0	

The Department of Forestry revised their thinning regimes for pine sawtimber again from 1990 (Kassier, 1991) and they had the following residual stems/ha:

Very good sites		*Normal sites*		*All sites*	
Age (yrs)	stems/ha	Age (yrs)	stems/ha	Age (yrs)	stems/ha
0	1372	0	1372	0	816
8	650	8	650	13	400
13	400	13	400	25+	0
18	250	25+	0		
25+	0				

Most softwood plantations grown primarily for pulpwood production are not thinned.

Common spacings used for pulpwood stands are:

(a) 2.4 x 2.4m (1736 stems/ha)

(b) 2.7 x 2.7m (1372 stems/ha)

(c) 3.3 x 3.3m (1111 stems/ha)

with rotations between 15 and 20 years.

Nelshoogte is described as a very good site and a value production regime with an initial planting density of 1372 stems/ha was derived from DMISER3 and compared with the above South African (SA) regime for a very good site:

Optimal regime from DMISER3

Age(yrs)	Remaining stems/ha
0	1372
5	1050
12	450
18	250
25-35	0

Figure 5.3 shows the saw-toothed pattern of basal area responses of the optimal regime and SA regime.

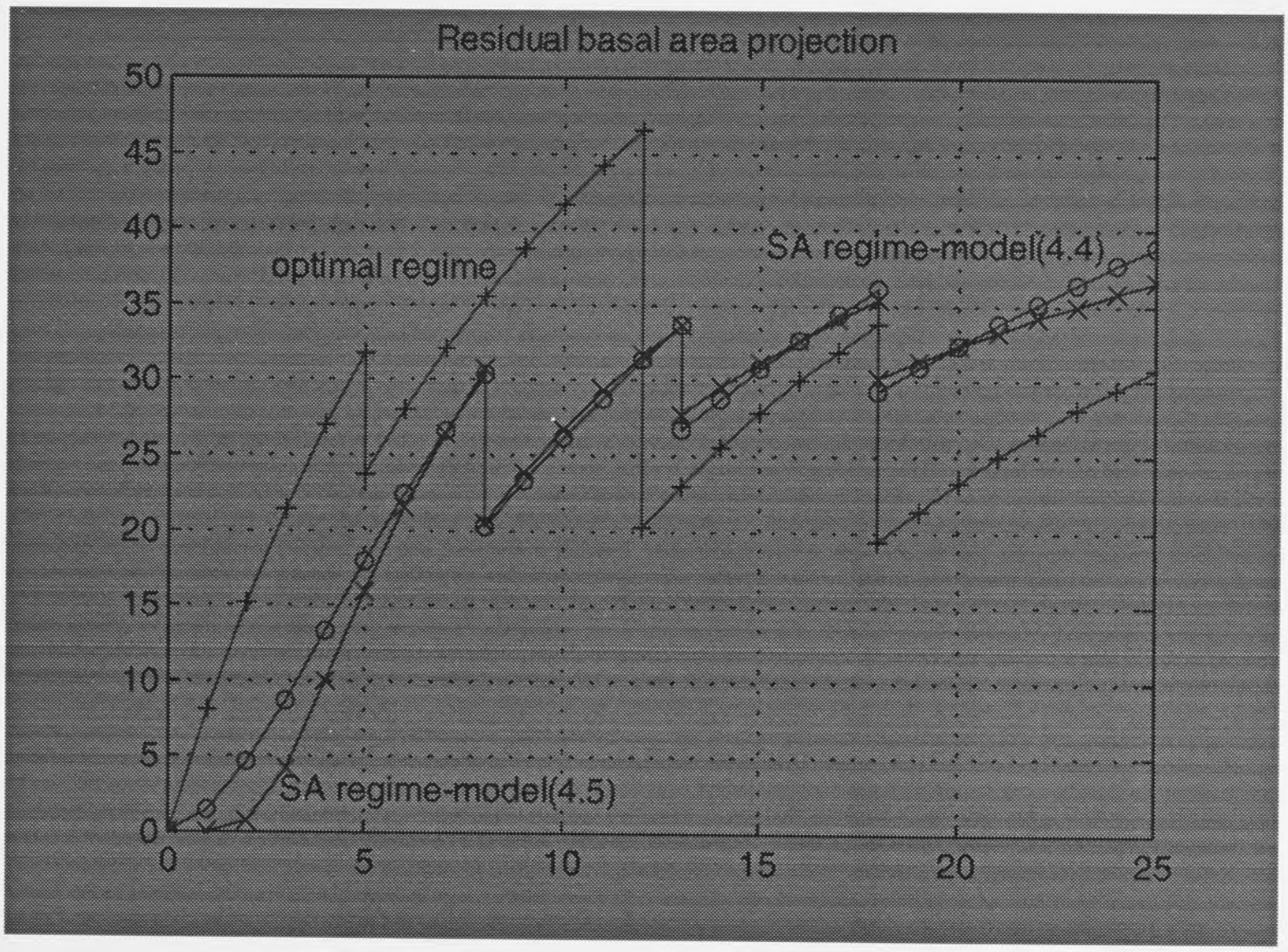


Figure 5.3: Residual basal area of the optimal regime and SA regime.

The basal area response for SA regime in Figure 5.3 was predicted using models (4.4) and (4.5), and the optimal regime was predicted using model (4.1). In the optimal regime, the first thinning is light, most likely non-commercial and is done early at age 5 years. The second thinning is delayed and followed by a heavy thinning. In the SA regime the first thinning is intermediate and it is delayed until age 8 years whereas the second thinning is a light one. After the second thinning a total of approximately 50 m^2/ha has been harvested from the optimal regime and 22 m^2/ha from the SA regime. The third thinning in both regimes occurs at the same time with the optimal regime having a harvest nearly as twice as that from the SA regime. If the final clearfell is carried out at age 25, the total harvest (including the harvest from the first, second and third thinning) from the optimal regime is 81 m^2/ha and 63-67 m^2/ha from the SA regime. As far as volume production is concerned, the optimal regime is certainly the better option but discussion of the value of the two regimes is inconclusive unless the benefits (of these two regimes) are translated into economic terms.

However, the main difference between the two regimes is that in the optimal regime the full growth vigour of the young trees is taken advantage of; a light first thinning to boost the growth followed by a heavy and delayed second thinning to capture the benefits of the first thinning on the residual trees. It is important to always thin at a time when the trees have the potential to respond quickly. The delayed second thinning in the optimal regime also ensures harvesting of bigger logs which would command a greater revenue. However, there are problems to watch for, when heavy thinnings are delayed:

(a) Extraction of bigger logs may necessitate use of heavy equipment which increases the likelihood of damage of the residual final crop;

(b) The incidence of wind damage to stands will increase following a heavy thinning on older stands; and

(c) There may be more small dimensioned logs (from the second thinning) which have high handling costs, although they can bring a useful early income with readily accessible markets.

Since the difference in basal area for the final crop is less than 10 m^2/ha between the SA and optimal regimes, both regimes would benefit from:

(a) Stumpage value increases as a stand gets older because the tree becomes bigger and more valuable products can be manufactured from larger timber;

(b) Logging costs per cubic metre that are lower the older and larger the trees being harvested are; and

(c) A large proportion of the wood in large logs has clear grain. Such quality differences usually cause the value of a stand of timber per cubic metre to rise as the trees grow bigger with age.

The control model thus has the ability to take advantage of the silvicultural benefits that are gained from thinning and delaying thinning at such times as to take full advantage of the trees when they are in the phase of responding with full growth vigour to the thinning treatment. As the trees grow older that full growth potential is lost, and the control model thus settles for moderate thinning.

Note that the benefits of a control model such as (5.3-6) are that as the stand conditions, climate, external economic forces and technology change, management will have the ability to simulate the constraints on such a control design and alter the regime(s) accordingly. It should be kept in mind that the best way to manage any system is flawless and consistent *information collection*, that always has to

be fed back into the control model. Forest managers should be kept abreast of the information that affects the productivity of the forest from forest inventories and political and economic trends so as to alter the forest management goals that are consequently translated into the control model.

5.5 Prediction and Partitioning of Yield

All diameter distribution yield systems predict the number of trees per hectare by diameter class and the average height for each diameter class. These data are then used in conjunction with an appropriate individual tree volume equation for calculation of the estimated per hectare yield (Clutter et al., 1983). Diameter distribution systems provide more detailed information for the stand structure.

The Weibull function can be used to predict the diameter distribution by employing DPPM. A first order stand mean diameter function was integrated with the Weibull probability density function, (5.20) to yield the following linear Weibull parameter estimation equations (Chikumbo, et al., 1992):

$$a(t+1) = \alpha a(t) + \beta \tag{5.17}$$

$$b(t+1) = \alpha b(t) \tag{5.18}$$

$$c(t+1) = c(t) \tag{5.19}$$

where

a = location parameter;

b = scale parameter;

c = shape parameter; and

α, β = estimated from the average diameter function.

A 3-parameter Weibull probability density function is used in the estimation process:

$$f(z) = \frac{c}{b}\left(\frac{z-a}{b}\right)^{c-1} exp\left(-\left(\frac{z-a}{b}\right)^{c}\right) \tag{5.20}$$

where

a, b, c = distribution parameters; and

z = random variable.

The first order ARX model for stand mean diameter for the Nelshoogte CCT trial was unstable and therefore not integrated with the Weibull probability density function. Assuming that a stand mean diameter model was available, it would have been possible to incorporate the Weibull distribution function in the control design. This would have assisted in determining the diameter classes at any one stage of the optimisation process. The Weibull parameters would be defined as states, represented by equations (5.18-20).

The Weibull has a cumulative distribution function (cdf) which would make it mathematically feasible to define the probability of a randomly selected variable (in this case, stand mean diameter). The standard input to DMISER3 (see appendix III) would therefore include the initial Weibull parameters, a, b and c (estimated from the initial diameter distribution). Since a and b from (5.17-19) will be dependent on density, there would be no reason for re-estimating the Weibull parameters after thinning. After an optimisation run, the DMISER3 output would look as follows:

STATE 1 stocking	STATE 2 basal area	STATE 3 height	STATE 4 avg diam	STATE 5 a	STATE 6 b	STATE 7 c
N_0	BA_0	Ht_0	dbh_0	a_0	b_0	c_0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
N_T	BA_T	Ht_T	dbh_T	a_T	b_T	c_T

The proportion of trees in a diameter class i where l and u define the upper and lower bounds of the class would be calculated as follows:

$$P_i\,(l<dbh_t<u) = exp[-\left(\frac{l_i - a}{b}\right)^c\,] - exp[-\left(\frac{u_i - a}{b}\right)^c\,] \qquad (5.21)$$

and the class frequency would be obtained as

$$\text{Class frequency} = N_t\,P_i \qquad (5.22)$$

If a single tree height and volume functions are available, class volumes can be obtained.

5.6 Conclusion

The control model (5.3-6) is simple and employs an automated process to determine optimal thinning strategies; this avoids the expensive and time consuming trials that are designed to find the appropriate strategies from the classical approach. In the event of market forces, political environment, climate and productivity change, there is a good chance of manipulating the constraints and/or the cost functional of the control model (5.3-6) and of redetermining an optimum strategy that can be implemented with confidence.

It has thus been demonstrated that growth functions developed as dynamical models can be easily formulated into a thinning optimisation problem which can be solved, using appropriate optimisation computer tools, to give optimal silvicultural management strategies. Note that the data used for the control model (5.3-6) came from one site and hence the results obtained are site specific. Availability of data from different sites will enable extension of this modelling approach beyond stand level.

Optimal control certainly has its place in forest management. A myriad of silvicultural questions at stand level can now be posed to provide more information for the foresters. The issues dealt with in this chapter were initial planting densities,

final crop density, timing of thinning, thinning intensity, number of thinnings per rotation period and rotation lengths.

Appendix 5A: A typical DMISER3 output for a volume production stand regime with a rotation age of 25 years and an initial planting density of 2000 stems/ha, where $0 < u(t) \leq 1000, \forall t \in [t, T]$.

The control sequence (i.e. the trees removed in thinning) at all ages are:

age	*control stems/ha*
0	0
1	0
2	0
3	0
4	0
5	0
6	60
7	207
8	266
9	198
10	29
11	62
12	39
13	53
14	26
15	10
16	32
17	17
18	23
19	554
20	14
21	13
22	12
23	11
24	38

The state variables at all ages are:

age	*remaining stems/ha*	*stand BA*	*stand height*
0	2000	0.0	0
1	2000	9.4	1.93
2	2000	17.3	5.13
3	2000	23.9	6.32
4	2000	29.5	8.37
5	2000	35.3	10.24
6	2000	38.2	11.91
7	1940	41.6	13.38
8	1733	45.4	15.7
9	1466	46.7	15.97
10	1268	48.6	17.3
11	1240	50.3	18.62
12	1178	51.8	19.77
13	1139	53.0	20.8
14	1086	55.2	21.71
15	1060	55.2	22.52
16	1051	56.0	23.23
17	1019	56.8	23.84
18	1002	57.5	25.39
19	980	58.2	25.87
20	426	58.7	25.3
21	411	58.5	26.36
22	398	58.2	27.31
23	386	57.9	28.16
24	376	57.6	28.92
25	338	57.3	29.6

The calculation of stand basal area is based on the first entry which does not reflrect thinning at time *t*. As a result the stand basal area sequence does not reflect the saw-toothed pattern.

CHAPTER 6

Recursive Identification for Forest Dynamical Models

Site as defined by the Society of American Foresters (1971), refers to '... an area considered in terms of its environment, particularly as this determines the type and quality of the vegetation the area can carry.' If required, site may be classified qualitatively into site *types*, by their climate, soil and vegetation, or quantitatively into site *classes*, by their potential to produce primary wood products. In so far as foresters are concerned, the primary purposes of site measurement are

(a) to identify the potential productivity of forest stands both present and future; and

(b) to provide a frame of reference for land management diagnosis and prescription.

Theoretically it should be possible to measure site directly by analysing the many factors affecting the productivity of forests, such as soil nutrients and moisture, temperature, available light, topography and so on. Although attempts at direct measurement of site have been made, such an approach may not be of immediate value to the practicing forester. Consequently, indirect estimates of site are frequently employed (Avery and Burkhart, 1994).

There are three indirect estimates of sites that are currently used, namely, tree height, physical-factors approach and indicator-plant approach. Avery and Burkhart (1994), give an overview of the three methods, but concentration will be on tree height in this chapter.

6.1 Mean dominant height

'Theoretically, height growth is sensitive to differences in site quality, little

affected by varying density levels and species compositions, relatively stable under varying thinning intensities and strongly correlated with volume.' (Avery and Burkhart, 1994). Other authors, such as Schmidt (1978) found that height growth was affected by stand density but to a lesser extent than diameter growth. No statistics were given to explain the extent of the effect. His findings, based on western larch (*Larix occidenlis* Nutt.), were that height growth is lost by crop trees growing in unthinned stands. In chapter four it was noted that *P. patula* was affected by stand density and that polymorphic behaviours of average stand height growth in the young stands were observed but converged to the same asymptote as the stand approached maturity.

A common measure of stand height is mean height defined either as the average of individual tree heights weighted by basal area or as the height of the tree of mean basal area obtained from a regression of height on basal area (Johnston et al., 1967). The problem with the use of mean height for classification of growth is that it is very sensitive to any form of thinning treatment or to natural mortality. Most thinning treatments tend to remove the shortest trees in the crop even if only to prevent death from suppression. Because of the different effects that different types of thinning have on mean height, various measures of height have been devised (such as mean dominant height) that are based primarily upon the dominant trees in the crop on the assumption that these are less likely to be removed in thinning and are less affected by density changes.

Mean dominant height (MDH) can still be affected by a type of thinning which removes trees in the larger size classes (crown thinning), although the proportion of such trees which can be removed is usually limited by the need to retain an adequate number of the more vigorous trees so as not to reduce increment per hectare. MDH based on say, the 20 largest trees per hectare is likely to be more

susceptible to the effects of crown thinning than a measure based on say, 100 trees per hectare.

It is therefore necessary to compromise between a measure based upon too small a proportion of the larger trees and one based upon too large a proportion, since the latter would be susceptible to low thinning as opposed to crown thinning. Also the compromise measure should be based on an index age so that consistency is maintained in describing site quality.

For the site index to be expressed on a standard basis, an index age must be assumed. In the USA, the period in the life of the stand that approximates the culmination of mean annual growth (tree size, be it volume or any size characteristic) in well-stocked stands, is usually selected as the index age. The average height of dominants and co-dominants measured at the index age is the site index. In most instances, however, stands measured are less than or greater than the index age. Consequently, a set of curves or an equation is needed to project the height to the standard reference age.

By using regression methods, a site index curve can be established and the most common transformation is

$$log\ H_d = b_0 + b_1 A^{-1} \tag{6.1}$$

where

H_d = height of the dominants and codominants (metres)

A = age of stand (years)

After the guide curve is estimated from model (6.1), an equation for site index as a function of measured age and height can be constructed by noting that when age is equal to index age A_i, height is equal to site index S, i.e.:

$$log\ S = b_0 + b_1 A_i^{-1} \tag{6.2}$$

This implies that

$$b_0 = \log S - b_1 A_i^{-1} \tag{6.3}$$

Substituting b_0 into model (6.1),

$$\log H_d = \log S - b_1 A_i^{-1} + b_1 A^{-1}$$

$$= \log S + b_1 (A^{-1} - A_i^{-1}) \tag{6.4}$$

Equation (6.4) can be used to generate site index curves and it can be used to estimate site index (height at index age) when age and height measurements are given. Other methods for site index construction include anamorphic and polymorphic methods (Avery and Burkhart, 1994).

Once site indices have been identified, growth functions, that are unique to each site, are developed. In an attempt to minimise the burden and cost of developing models for each site quality, site index has provided over the years a means of adapting a model structure to different sites. A BA function such as (2.19) has site index, S (mean dominant height at age 20) as one of its variables and by varying S, equation (2.19) is supposed to give a BA response at specified sites. The success of this approach depends on:

(a) the accuracy of the site index estimation;

(b) the response consistency of the stochastic process, i.e. the BA response of a forest system driven by a stochastic process that can vary from site to site; and

(c) the ability of a model structure to capture the complexity of the observed phenomenon from all the sites in consideration.

Dynamical models can be directly calibrated to account for different productivity levels for different sites without necessarily defining a site index. A BA function for say site *A* can be calibrated to suit site *B*, provided at least some BA re-

measurement data are available from site *B*. To have confidence in the model, a recursive identification process may be set up such that when new data from site *B* are available, they can be used to fine tune the 'guessed' model. In general terms, recursive identification involves the construction of a dynamical model as a separate procedure, from a batch of measured data collected from forest site *B*, where the whole batch is available at all stages of the procedure for forest site *A* (Ljung and Soderstrom, 1983).

6.2 Recursive identification

Assuming the BA model (4.1) of chapter four, $BA(t) = aBA(t - 1) + b$ at time t, has been calibrated (from its family of curves) to fit an observed response sequence $\{BA\}$ over some time window from a different site, it should be possible to fine tune a and b with newly available data. Mathematically, the ultimate $\hat{a}(t)$ and $\hat{b}(t)$ after t data points are collected must be chosen such that

$$BA(l) = a(t)BA(l - 1) + b(t) \tag{6.5}$$

for all integers l in $[1, t]$.

Assuming $t = 3$ i.e. $l = 1, 2,$ and $3,$

$$BA(1) = \hat{a}(3)\, BA(0) + \hat{b}(3) = [BA(1)\ 1]\begin{bmatrix}\hat{a}(3)\\ \hat{b}(3)\end{bmatrix} \tag{6.6}$$

$$BA(2) = \hat{a}(3)\, BA(1) + \hat{b}(3) = [BA(2)\ 1]\begin{bmatrix}\hat{a}(3)\\ \hat{b}(3)\end{bmatrix} \tag{6.7}$$

$$BA(3) = \hat{a}(3)\, BA(2) + \hat{b}(3) = [BA(3)\ 1]\begin{bmatrix}\hat{a}(3)\\ \hat{b}(3)\end{bmatrix} \tag{6.8}$$

These 3 simultaneous equations can be combined into one matrix equation:

$$\begin{bmatrix} BA(0) & 1\\ BA(1) & 1\\ BA(2) & 1\end{bmatrix}\begin{bmatrix}\hat{a}(3)\\ \hat{b}(3)\end{bmatrix} = \begin{bmatrix}BA(1)\\ BA(2)\\ BA(3)\end{bmatrix} \tag{6.9}$$

The matrix can be generalised as

$$\begin{bmatrix} BA(0) & 1 \\ . & . \\ . & . \\ . & . \\ BA(t-2) & 1 \\ BA(t-1) & 1 \end{bmatrix} \begin{bmatrix} \hat{a}(t) \\ \hat{b}(t) \end{bmatrix} = \begin{bmatrix} BA(1) \\ . \\ . \\ . \\ BA(t-1) \\ BA(t) \end{bmatrix} \tag{6.10}$$

By appropriate definition of $\boldsymbol{X}_t$, $\boldsymbol{BA}_t$, and $\hat{\theta}(t)$ (6.7) and (6.8) can be compactly written as

$$\boldsymbol{X}_t \ \hat{\theta}(t) = \boldsymbol{BA}_t \tag{6.11}$$

Note that the subscript t on $\boldsymbol{X}_t$ and $\boldsymbol{BA}_t$ indicates their row dimension. In practice no single choice for $\hat{\theta}$ exists such that (6.11) is an equality, due to the unavoidable presence of noise in measuring the *BA* or the fact that any forest growth pattern is not actually linear and time invariant as is the difference equation in (6.5). In other words, even for the 'best' $\hat{\theta}$, (6.11) is an approximation of

$$\boldsymbol{X}_t \ \hat{\theta}(t) = \boldsymbol{BA}_t + \boldsymbol{E}_t \tag{6.12}$$

where $\boldsymbol{E}_t$ is the vector of 'unremovable' residual errors.

Thus it is reasonable to collect data over a time window of length t in order to effectively 'average' over more of the 'unremovable' residuals in the solution of (6.12). Several recursive identification algorithms, can be applied from this point, such as the recursive least-squares algorithm (RLS), the recursive weighted least-squares algorithm (RWLS), least mean squares algorithm (LMS) and normalised least mean squares (NLMS) (Johnson, 1988).

When the output *BA* in (6.10) is measured in the presence of noise, the noiseless *BA* in (6.5) is different from the noisy *BA* in $\boldsymbol{X}_t$ and $\boldsymbol{E}_t$ in (6.12). In fact, $\boldsymbol{E}_t$ becomes correlated, in a complicated manner, with $\boldsymbol{X}_t$ even if this measurement noise

is zero-mean and uncorrelated with the input. This leads to a bias in solution for all the algorithms since they all minimise the squared prediction error (Johnson, 1988).

The following forestry example illustrates a situation where calibration of a growth model followed by recursive identification (to fine tune the 'guessed' model) was used.

6.3 Forestry example of recursive identification: development of a generic yield curve

Generic forest yield models, covering all possible growing sites and silvicultural regimes and several tree species, are simply non-existent. Yield is the amount of volume per unit area available for harvesting at a given time. Thus yield can be regarded as the summation of the annual increments. Models, however, are required for yield projections. For the Victorian hardwood forests, the calculation of yield is a requirement under the Forest Act 1958, for each Forest Management Area (FMA).

In this example, a volume function from a locality (Bruces Ck) in the Central Highlands was developed from the available data using system identification, and the mean annual increment (mai) data were derived from it. MAI is calculated from stand volume prediction by dividing the stand volume at time t by the age (t). Because of the closeness of the productivity of the other localities, recursive identification was used to estimate the volume functions of these localities that had very little data. As more re-measurement data become available from these localities, the models will be further adjusted until there are no appreciable changes in the parameters i.e., until the expected trends are fully described by the models. The mai data (from Bruces Ck) were used to model an mai curve and from it a family of mai curves were generated and used to estimate mai models for a different region (Midlands).

Data

Data from the mixed species forest came from a total of twenty-six plots. Ten plot datasets were from four localities in the Central Highlands region of Victoria called Loop Rd, Acheron, Black Range and Bruces Ck. Datasets for sixteen other plots came from Wombat State Forest in the Midlands region. Table 6.1 shows a summary of the plot data including the species from the ten plots in the Central Highlands and the sixteen plots from the Wombat forest. All the Central Highlands plots were untreated with the exception of Bruces Ck where the plots had been thinned. The continuous forest inventory plots from the Wombat forest have not been treated in the past thirty years although evidence from stumps seem to suggest that some harvesting was done fifty years ago. The data consisted of age, volume (total gross bole volume under bark - tvub) and mai (of tvub). Volume (under bark) was calculated by product classes for each tree in the continuous inventory plots in the Wombat forest, namely stump volume, sawlog volume and residual roundwood volume.

Table 6.1: Summary of the plot data from the Central Highlands and Midlands regions. Age, volumes and mai's are expressed as minimum-mean-maximum values. The codes used for the different species are as follows: EO-*Eucalyptus obliqua*; ER-*Eucalyptus radiata*; EF-*Eucalyptus fastigata*; ES-*Eucalyptus sieberi*; ERu-*Eucalyptus rubida*; ED-*Eucalyptus dives*; ECi-*Eucalyptus cinerea*; EC-*Eucalyptus cypellocarpa*; EV-*Eucalyptus viminalis*; and EOv-*Eucalyptus ovata*.

	SPECIES	*NUMBER OF PLOTS*	*AGE*	*TVUB*	*MAI(TUB)*	*SAWLOG VOL*	*MAI(SAW-LOG VOL)*
Acheron	EO, ER	3	27-31-37	262-318-411	9.7-10-11		
Black Range	EO,ER	2	24-25-27	233-256-286	9.7-10-10.7		
Bruces Ck	EO,EF	3	42-56-73	364-866-1531	8.7-15-21		
Loop Rd	EO,ES	2	17-21-25	155-217-288	8.7-10-11.8		
Wombat Forest	EO,ERu,ER, ED,ECi,EC, EV,EOv	16	47-80-105	280-675-1552	4-8.3-16.4	5-112-311	0.1-1.4-3.1

Derivation of Volume and mai functions

The volumes from the four localities showed linear trends which did not have enough response information that would lead to a model estimation of maximum mai at some time t. Only the tvub's from one plot at Bruces Ck showed a linear trend that was just starting to converge towards an asymptote. Natural logarithms were used to enhance the asymptotic convergence of the volume response. Using the plot data from Bruces Ck, the mai function was developed as follows:

(a) a first order dynamical model was developed for volume estimation;

(b) using the function from (a), a simulation was done over a 0-200 year period (which showed the complete sigmoidal curve);

(c) the predicted volume from (b) was used to determine mai; and

(d) the predicted mai from (c) was used to develop a second order dynamical model for mai (of tvub).

The volume function had the following form:

$$y(t) = ay(t-1) + b \tag{6.13}$$

where

y = log(tvub (m^3/ha))

t = time (years);

a = 0.9715;

b = 0.2256.

Model (6.13) had a low loss function of *2.24* and Akaike's Final Prediction-error (FPE) of 2.4 m^3/ha, i.e. the variance, on average, that will be obtained when a model is applied as a predictor to an independent data set. The mai curve was as follows:

$$y(t) = a_1y(t-1) - a_2y(t-2) + b \tag{6.14}$$

where

y = mai (m^3/ha/year);

$a_1 = 1.9721;$

$a_2 = 0.973;$

$b = 0.0134,$

with a loss function 0.086 and Akaikes FPE of 0.089 $m^3/ha/year$.

For the other localities in the Central Highlands, Acheron and Loop Rd, adaptive parameter estimation techniques were used to develop their volume functions. The following volume equations were obtained for Loop Rd and Acheron localities respectively, by using NLMS with equation (6.13) as the starting model and for single plots in each locality;

$$y(t) = 0.9698y(t-1) + 0.2253 \quad (6.15)$$

$$y(t) = 0.9694y(t-1) + 0.2252 \quad (6.16)$$

where

$y = \log(\text{volume - } m^3/ha)$

The Black Range data had only three time intervals, and were therefore not included in this analysis.

In order to have an mai curve to use for the Wombat forest (in the Midlands region), the mai model (6.14) was calibrated to the rest of the mai data from the Wombat forest by varying the initialisation and parameter b simultaneously. This approach was also used to fit the sawlog mai data. A suite of curves is shown in Figures 6.1-2 for tvub mai and sawlog mai respectively and the suit of curves tends to envelope most of the data.

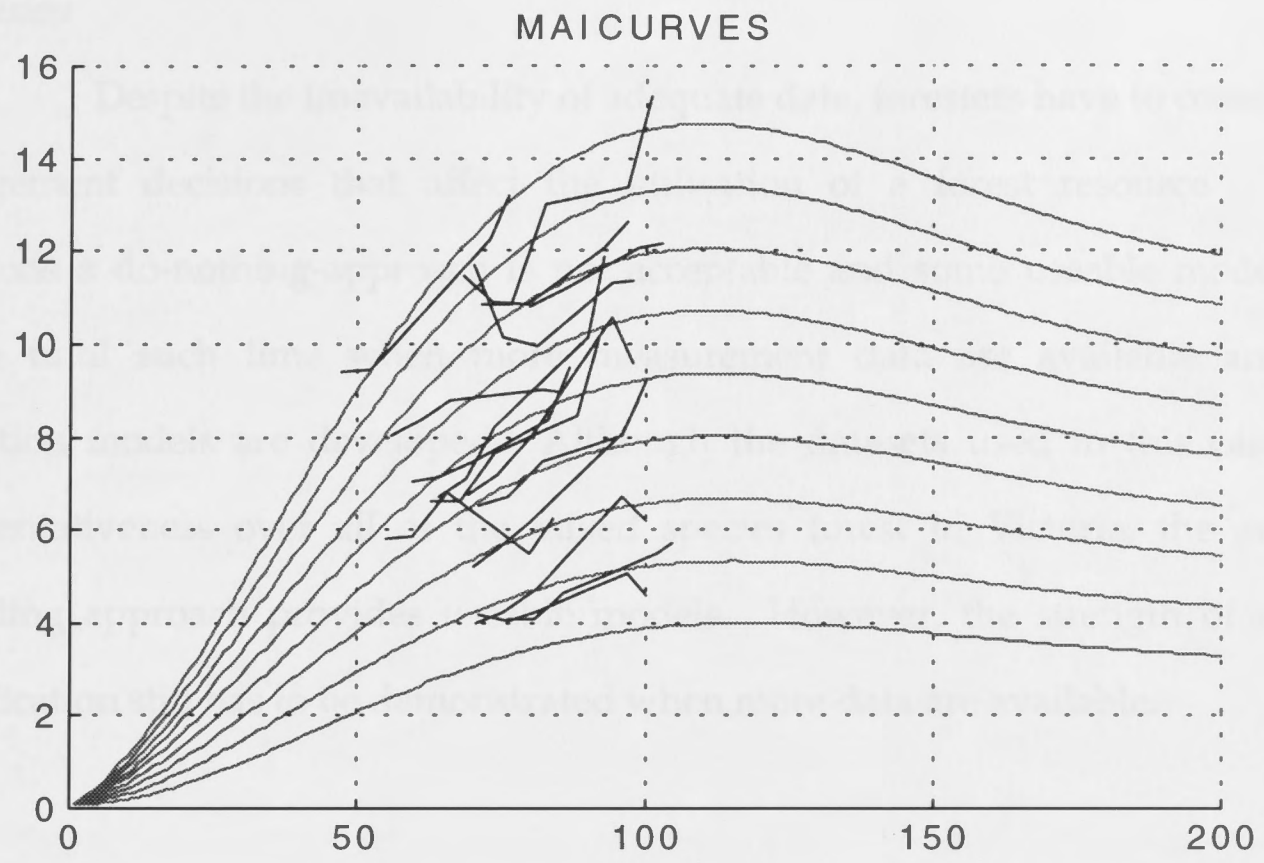


Figure 6.1: A suit of tvub mai curves derived from the mai function (6.14) with the Wombat forest plot data superimposed. Initialisation range for mai was 1-1.8 at a fixed interval of 0.02 and the parameter b from the mai function had a range of 0.003-0.011 at a fixed interval of 0.001.

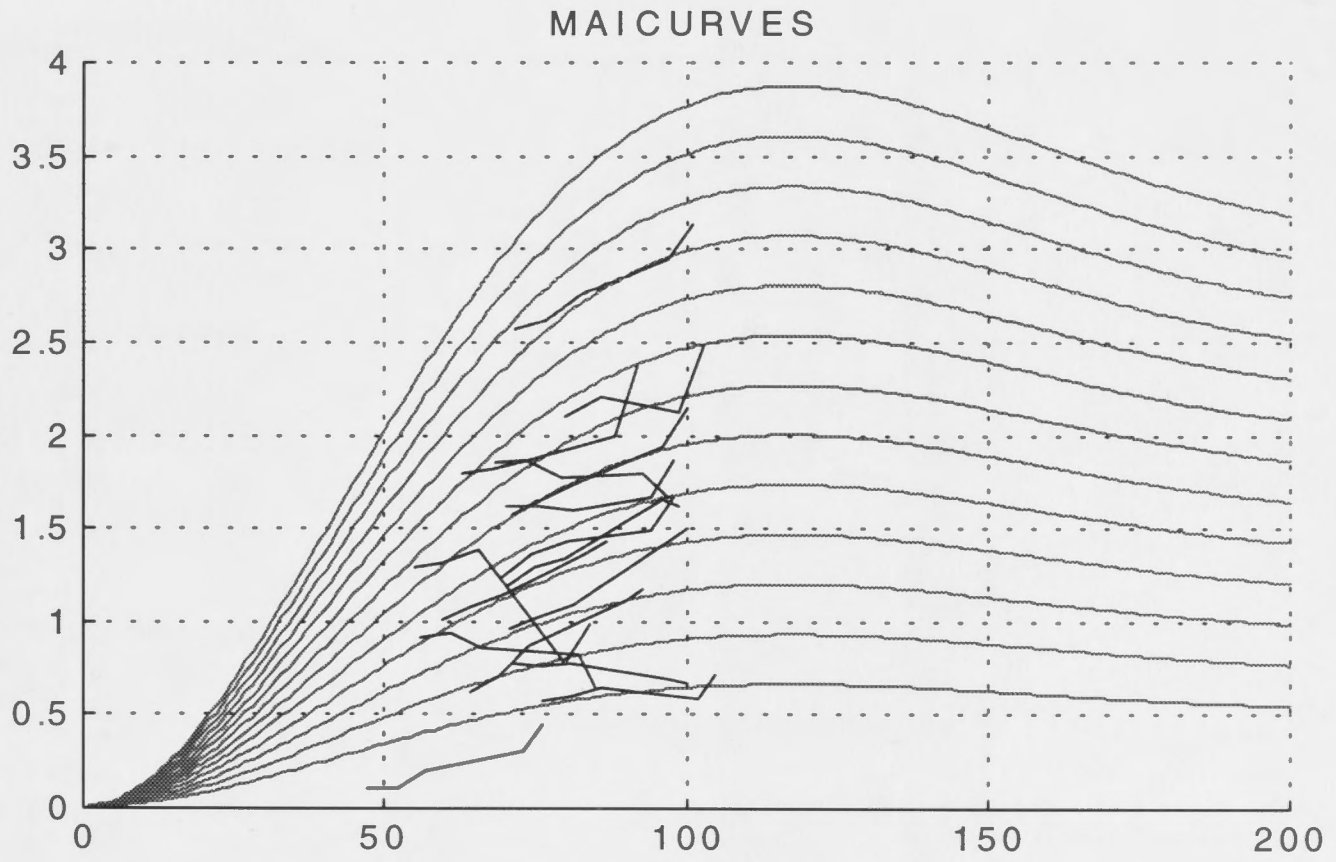


Figure 6.2: A suit of sawlog mai curves derived from the mai function (6.14) with the Wombat forest plot data superimposed. Initialisation range for mai was 1-1.5 at a fixed interval of 0.02 and the parameter b from the mai function had a range of 0.0005-0.003 at a fixed interval of 0.0002.

Conclusion

Despite the unavailability of adequate data, foresters have to come up with management decisions that affect the utilisation of a forest resource. In most situations a do-nothing-approach is not acceptable and some useable model would suffice until such time when more measurement data are available and better prediction models are developed. Although the datasets used in this case lacked representativeness over all of the mixed species forest in Victoria, the advocated modelling approach provides useable models. However, the strength of recursive identification still has to be demonstrated when more data are available.

CHAPTER 7

Extensions

7.1 Hydrological applications

Currently there is some work being done in some of the Victorian State (Australia) water supply catchment areas, that involves the prediction of possible water yield fluctuations, following a bushfire period in mixed species eucalyptus forests. The idea is to extrapolate the results from this investigation to other catchment areas, where there is forestry activity (timber production), so that the utilisation of this resource does not interfere with the water yields. Any major adverse effects from this utilisation would mean changes in the forest management practice.

The large catchment areas, Warburton, Neerim South, Thompson and Maroondah, with monthly stream run-off data (measured over the past 100 years) revealed no yield effects following a fire in an established mixed species forest. System identification was used to model the run-off data that were identified as white noise despite an initial treatment of the data with Fourier transforms. These transforms are commonly used to find the frequency components of any signal (response) buried in a noisy time domain data sequence. The results of this analysis were in agreement with the independent findings by Kuczera (1985).

7.2 Economic improvements

There is still a wide scope for improvement of the control design in chapter five. For example, a suitable aggregate measure for various economic inputs can be developed so that an economic component can be built into the control design. Research in this area would enable foresters to account for the overall cost structure of the final forest products. A starting point would be calculations of shadow prices for the different forest products, i.e. residual roundwood, small sawlogs, big sawlogs and

veneer logs which are categorised by end diameter size under bark. A shadow price is an estimated or hypothetical price used in economic analyses. It is derived by adjusting the actual prices to account for price deviations caused by market distortions which result in the reigning market prices not reflecting the true social value of the product under conditions of perfect resource allocation (Byron and Douglas, 1981). For example, the shadow price of residual roundwood on the stump can be calculated as follows (Byron and Douglas, 1981):

$$\text{SRW} = \frac{(F + f_c + W_c + s - PC)}{R_c} - L \tag{7.1}$$

where

SRW = shadow price of residual roundwood

F_c = price of pulp

f_c = freight and insurance costs

W_c = cost of wharf handling, etc

s = freight cost of transporting from port to mill

PC = manufacturing costs

R_c = pulp recovery

L = logging costs.

The optimal control criteria then takes the form (see equation (5.2)):

$$J = \max_{u(t)} \sum_{t=1}^{T} SRW(t)[\frac{u(t)}{x(t)} V_t(x(t))] \tag{7.2}$$

where

SRW(t) = shadow price of residual roundwood at time, *t*.

Alternatively, *stumpage* can be included in the control design as a state. Stumpage value is the maximum price that competitive buyers would be prepared to pay for the timber standing in the forest. It follows that the stumpage value is equal to the revenue that an efficient producer could expect from harvesting the timber and selling it in the best available market, less the expected costs in harvesting the timber and delivering it to the market. A forest plantation increases in stumpage value with time and follows a sigmoid curve, its slope increasing up to an inflection point then decreasing (Pearse, 1990). A delayed first order or a second order dynamical function can be used to identify the sigmoidal behaviour. The reason for this increasing trend is that stumpage is a time function dependent on other variables shown below:

$$S = f(BA, H, H_c, Y_c, L_c) \tag{7.3}$$

where

S = stumpage value of forest land

BA = basal area - m^2/ ha

H = height - m (BA and H define the size of the log)

H_c = harvesting costs which are lower for bigger logs

Y_c = yarding costs which are less per m^3 for larger pieces of logs

L_c = logging costs which are normally lower per m^3 for the larger logs

Given that the Weibull parameters are also defined as states for the control model, a good prognosis of the total volume and log distribution can be obtained (Eriksson and Sallnas, 1987).

Once a log distribution is obtained, it is not difficult to construct via dynamic programming methods, a log cutting algorithm that maximises return for any given stem size. However, more information is required that includes a stem's complete geometric profile (length, diameter, taper and curvature), quality variations along the stem (including knots and rot), and varying economic utilities of potential

logs of different geometries and qualities in different mills. Indeed various versions of such a dynamic programming algorithm have been discussed in literature, starting with Pnevmaticos and Mann (1972). The Weyerhaeuser Company in Tacoma, Washington (USA) developed a Decision Simulator (DS) that allows users to test different crosscutting and log-allocating decisions for economic value, interactively. The system simulates all salient features of the actual decision-making situation, while providing a private environment for 'non-destructive' learning by doing. The operator sees on a video display a realistic representation of each stem. The DS permits the operator to inspect the stem (for example, roll and rotate it) and then cut and allocate it. The operator sees the logs and the profit contribution from his decisions for each stem and also sees the stem cut and allocated by the dynamic programming algorithm to maximise profit contribution (Lembersky and Chi, 1984). The operator can recut the same stem repeatedly to explore alternate decisions.

7.3 Speculative applications

Further developments of the control design problem in thinning strategies rests in the work covered in chapter six, i.e. adapting the process models to different sites by recursing and developing the controllers. At this stage it can almost be concluded that growth responses in intensively managed plantations can often be represented by first order dynamical models and thus optimisation for stand density is basically the control of first order models. Application of specific controllers for specific forest plantations will yield different control sequences. It would be worth investigating whether any similarities would occur in the control sequences generated such as, when, in the rotation period (some fraction of time of the rotation length) should a thinning occur and if it does, what percentage of the number of trees is thinned.

If a consistent pattern is observed, be it for maximising volume production or net present value, it would then be possible to formulate a generalised optimal thinning strategy for intensively managed forests. This strategy can then be applied by a field forester without having to understand and directly apply the mathematics of multistage optimisation. Problems with such an approach may arise for companies with large forest resources, given the erratic fluctuations of demand and supply on the markets, i.e. a constant monitoring of plantation management strategies is required so that a fairly good level of flexibility is kept to accommodate the changing markets.

CHAPTER 8

Conclusion

A Russian by the name of Pontryagin, in 1956 (Pontryagin et al., 1962), developed an efficient solution technique called maximum principle. The American operations research scientist, Richard Bellman (1957), developed the concept of dynamic programming over a variety of practical problems that required sequential solutions for optimal decision-making. Bellman was exceptional in that he could demonstrate his concepts on dynamic programming by looking at everyday problems that managers from different disciplines encountered. As a result he managed to successfully sell his brilliant ideas.

Further developments and major breakthroughs came with Rudolf Kalman (1960) with the concepts on controllability and observability of systems. These concepts answered the questions on how to steer a dynamical system from a given initial state to any other state and how to determine the state of a dynamical system from observations of inputs and outputs. This knowledge was also applicable to multiple-input/multiple-output systems.

Since then the bulk of control systems in engineering, both analog and digital, have been based on state space models derived from the physics of their systems or derived from their input/output models, where dynamic programming or maximum principle were used to find solutions of the controller. This wealth of analytical techniques provided easier methods for solving control problems. Since the concepts are application independent, foresters made their first attempt in the late sixties to tap into this wealth of knowledge and apply it to various forest management problems. Success was made in other areas such as log assortment but optimising for a thinning schedule proved to be a difficult problem for years.

It has been twenty-seven years since the first attempt was made to use dynamic programming (Amidon and Akin, 1968) to optimise a thinning schedule over an entire rotation. Explicit derivation of solution procedures were unclear throughout the seventies and eighties and Chen et al., (1980) expounded on the problems that were attributed to this. The problems were mainly, a lack of clarity about the conditions that had to be met for optimality and hence application of the solution procedures, and the unavailability of appropriate forest growth models that related directly to the decision variable.

By strictly adhering to the requirements of an optimal control problem, it has been possible, in this research work, to develop the appropriate functions via system identification. Nonlinear basal area and height functions were developed with a linear time-invariant model structure governed by parameters that were functions of initial stand density. This made the BA function responsive to thinning. The control model was defined with a state vector of stand density (stems/ha), stand basal area (m^2/ha) and average stand height (m). The output of the control model was a combination of the state vectors to predict stand volume. The input or control variable expressed in stems/ha, determined the response of the control model. A cost functional for volume production or value production was formulated in a dynamic programming framework and the solution determined using iterative solution techniques for optimisation (i.e. dynamic programming or maximum principle).

References

Amidon, E. L. and G. S. Akin. (1968). "Dynamic programming to determine optimal level of growing stock." **For. Sci. 14**: 287-291,

Anderson D.J. and B.B. Bare (1994). "A dynamic programming algorithm for optimisation of uneven-aged forest stands". **Can. J. For. Res. 24**: 1758-1765,

Anderson, B. D. O. and J. B. Moore. (1990). Optimal Control: Linear Quadratic Methods. Prentice Hall Information and System Sciences Series. New Jersey, Prentice Hall Inc.,

Arimizu, T. (1958). "Regulation of the cut by dynamic programming." **J. Operations Res. Soc. Japan. 1(4)**: 175-182,

Arthaud, G.J. and W.D. Klemperer. (1988). "Optimising high and low thinnings in loblolly pine with dynamic programming." **Can. J. For. Res. 18**: 1118-1122,

Arthaud, G.J. and D.B. Warnell. (1994). A comparison of foward-recursion dynamic programming and A* in forest stand optimisation. In: Management Systems for a Global Economy with Global Resource Concerns, Edited by Sessions, J. and Brodie, J.D., Pacific Grove, Carlifornia,

Astrom, K. J. and B. Wittenmark. (1984). Computer controlled systems : theory and design. Englewood Cliffs N.J., Prentice-Hall,

Athans, M. and P. L. Falb. (1966). Optimal control : an introduction to the theory and its applications. N.Y., McGraw-Hill,

Avery, T. E. and H. E. Burkhart. (1994). Forest Measurements. Forest Resources. New York, McGraw-Hill,

Bailey, R. L. and J. L. Clutter. (1974). "Base-age invariant polymorphic site curves." **For. Sci. 20**: 155-159,

Barlow, N.D. (1983). The role of modelling in practical pest management. In: Cameron, P.J., Wearing, C.H., and Kain, W.M. (Eds) Proceedings of Australasian Workshop on Development and Implementation of IPM. Government Printer, Auckland, New Zealand,

Barnett, S. and R. G. Cameron. (1985). Introduction to Mathematical Control Theory. Oxford Applied Mathematics and Computing Science. New York, Oxford University Press,

Bellman, R. (1961). Adaptive Control Processes. New Jersey, Princeton Univ. Press,

Bellman, R. and S. E. Dreyfus. (1962). Applied Dynamic Programming. New Jersey, Princeton Univ. Press,

Bellman, R. E. (1957). Dynamic programming. Princeton, NJ, Princeton Univ. Press,

Bertsekas, D.P. (1976). Dynamic Programming and Stochastic Control, Academic Press, New York,

Bertalanffy, L. v., (1973). General system theory : foundations, development, applications. New York, G. Braziller,

Boltiyanskii, V. G., R. V. Gamkrelidze and L. S. Pontryagin. (1956). On the Theory of Optimal Processes. Doklady Akad. Nauk SSSR, (in Narhorski et al., 1983),

Boltyanskii, V. G. (1956). The Maximum Principle in the Theory of Optimum Processes. Doklady Akad. Nauk SSSR, (in Nahorski et al., 1983).

Boltyanskii, V. G. (1971). Mathematical methods of optimal control. Balskrishnan-Neustadt. New York,

Borowski, E.J. and Borwein, J.M. (1989). Dictionary of Mathematics. Harper Collins Publishers. Greta Britain.

Brack, C. L. (1985). The RADHOP System. In: Modelling Trees, Stands and Forests. Edited by J. W. Leech, R. E. McMurtie, P. W. West, R. D. Spencer and B. M. Spencer, University of Melbourne, Melbourne, Bulletin No.5. 509-525,

Bredenkamp, B.V. (1993). The volume of standing trees. In: Van der Sijde, H.A. (ed), South African Forestry Handbook. South African Institute of Forestry. Pretoria. 846p,

Bredenkamp, B.V., (1984). The CCT concept in spacing research -A review. In Proc. IUFRO symposium on site and productivity of fast growing plantations. Pretoria and Pietermaritzburg, South Africa. 313-332,

Bredenkamp, B.V., Venter, J.S.J. and Haigh, H. (1983). "Early Respacement and fewer thinnings can increase profitability of coniferous sawtimber production." **S. Afr. For. J. 124**: 36-42,

Brillinger, D. R. (1974). Time series: data analysis and theory. New York, Holt, Rinehart and Winston,

Brodie, J. D., D. M. Adams and C. Kao. (1978). "Analysis of economic impacts on thinning and rotation for Douglas-fir using Dynamic Programming." **For. Sci. 24**: 513-522,

Buongiorno, J. and J. K. Gilless. (1987). Forest Management and Economics. New York, NY 10022, Macmillan Publishing Company,

Burges, D. and Graham, A. (1980) Introduction to Control Theory Including Optimal Control. Ellis Horwood Limited, Chichester, Britain,

Byron, R.N. and Douglas, J.J. (1981). Log Pricing in Australia: Policies, practices and consequences. BFE Press, Canberra, Australia,

Chang, S. S. L. (1960). Digitized Maximum Principle. Proc. IRE. pp 2030-2031, (In: Nahorski et al., 1983),

Chang, S. S. L. (1961). Synthesis of Optimum Control Systems. New York, McGraw-Hill,

Chen, C. M., D. W. Rose and R. A. Leary. (1980). "How to Formulate and solve optimal stand density over time problem for even-aged stands using dynamic programming." **General Technical Report NC-56**: 17,

Chikumbo, O. (1991). Characterising Diameter Distribution of Monterey Pine and Growth Prediction Based on Dynamical Models. Honours Thesis. Australian National University,

Chikumbo, O., R. N. James, et al. (1993). Mortality simulations in Pinus radiata plantations in the Tarawera valley regimes trial. Modelling and Simulation of Environmental Changes. Edited by Jakeman, A and McLeer, M., University of Western Australia, Perth. 869-874,

Chikumbo, O. and I. M. Y. Mareels (1993). A simulation model for diameter growth in Pinus radiata plantations. Modelling and Simulation of Environmental Changes. Edited by Jakeman, A. and McLeer, M., University of Western Australia, Perth. 851-856,

Chikumbo, O. and I. M. Y. Mareels. (1995a). Dynamical models for ecological/ environmental modelling and planning. In: International Congress on Modelling and Simulation Proceedings. Edited by Binning, P., Bridgman, H., and Williams, B. The University of Newcastle, Newcastle, NSW. vol (1): 239-245,

Chikumbo, O. and I. M. Y. Mareels. (1995b). Optimal thinning strategies based on dynamical models and the maximum principle. In: Applicaction of New Technologies in Forestry. Edited by L. Bren and C. Greenwood. Ballarat, Victoria. Institute of Foresters of Australia. vol (1): 239-245,

Chikumbo, O., I. M. Y. Mareels and B. J. Turner. (1992). Integrating the Weibull into a dynamical model to predict future diameter distributions. Edited by G. B. Wood and B. J. Turner. Integrating Forest Information over space and Time. Australian National University. ANUTECH Pty Ltd. 94-102,

Clark, C. W. (1976). Mathematical Bioeconomics: The Optimal Management of Renewable Resources. New York, Wiley - Interscience,

Clutter, J.L., Fortson, J.C., Pienaar, L.V., Brister, G.H. and Bailey R.L. (1983). Timber Management: A quantitative Approach. John Wiley & Sons, Inc. New York,

Commonweath of Auatralia, (1993). Biodiversity: the role of protected areas. Report of the House of Representatives, Standing Committee on Environment, Recreation and the Arts. Australian Government Publishing Service, Canberra,

Cuff, W. and Baskerville, G. (1983). Ecological modelling and management of spruce budworm infected fir-spruce forests of New Brunswick, Canada. In: Lauenroth, W.K., Skogerboe, G.V. and Flug, M (Eds) Analysis of Ecological Systems: State-of-the-art in Ecological Modelling, Elsevier,

Daintith, J. and R. D. Nelson. (1989). Dictionary of Mathematics. England, Penguin Books Ltd,

Dargavel, J. (1992). Centre for Resource & Environmental Studies (CRES) - ANU (Canberra),

de la Barriere, R. P. (1967). Optimal Control Theory; A Course in Automatic Control Theory. West Washington Square, Philadelphia, W.B. Saunders Company, (Translated from French by Scripta Technica and edited by B.R. Gelbaum),

Draper, N. and Smith, H. (1981) Applied Regression Analysis. Second Edition., John Wiley & Sons, USA,

Elgerd, O. I. (1967). Control Systems Theory. Tokyo, McGraw-Hill Kogakusha,

Eriksson, L.O. and Sallnas, O. (1987) "A model for predicting log yield from stand characteristics." **Scand. J. For. Res. 2**:253-261,

Fan, L. T. and C. S. Wang. (1964). The Discrete Maximum Principle. New York, Wiley,

Filius, A. M. and M. T. Dul. (1992). "Dependence of rotation and thinning regime on economic factors and silvicultural constraints: results of an application of dynamic programming." **Forest Ecology and Management. 48**: 345-356,

Fogiel, M. (1987). The Essentials of Automatic Control Systems/Robotics II. Piscataway, New Jersey 08854, Research and Education Association,

FORPLAN Version 2: Users Guide. (1991). Land Management Planning Systems Section, United States Department of Agriculture, Forest Service, Washington, DC.

Gamkrelidze, G. V. (1957). "On the Theory of Optimal Processes in Linear Systems." **Doklady Akad. Nauk SSSR. 116**(1): (in Nahorski et al., 1983),

Gamkrelidze, G. V. (1958a). "The Theory of Time Optimal Processes in Linear Systems." **Izv. Akad. Nauk SSSR, Ser. Matem. 22**(4): English translation in Report N0. 61-7, Univ. of California, Department of Engineering, Los Angeles, California. (in Nahorski et al., 1983),

Gamkrelidze, G. V. (1958b). "On the General Theory of Optimum Processes." **Doklady Akad. Nauk SSSR. 123**(2): (English translation in Automation Express, Vol 1, 37-39, 1959),

Garcia, O. (1984). "Models and Estimation procedures." **For. Sci. 14(1)**: 65-88,

Garcia, O. (1988). "A (Re)view - Growth Modelling." **NZ Forestry. 33(3)**: 14-17,

Goodwin, G. C. and K. S. Sin. (1984). Adaptive filtering prediction and control. Englewood Cliffs, N.J., Prentice-Hall,

Haight, R. G., J. D. Brodie and W. G. Dahms. (1985). "A dynamic programming algorithm for optimization of lodgepole pine management." **For. Sci. 31:2**: pp 321-330,

Halkin, H. (1964). "Optimal Control for Systems Described by Difference Equations." In: Advances in Control Systems, Edited by Leondes, C.T. (1964), Vol. 1, Academic Press,

Hann, D.W. and Brodie, J.D. (1980) Even-Aged Management: Basic Managerial Questions and Available or Potential Techniques for Answering Them. General Technology, INT-83. USDA Forest Service, Intermountain Forest and Range Experiment Station, Ogden, Utah. 29 pp,

Harrison, W. M., L. V. Pienaar and H. Kotze. (1994). Planting density effects on growth and yield in the Nelshoogte correlated curve trend experiment with *Pinus patula*. 15 December 1994, FOR-C-264, The Division of Forest Science and Technology, CSIR,

Holtzman, J. M. (1966). "Convexity and the Maximum Principle for Discrete Systems." **IEEE Trans. Autom. Control. AC-11**(1): 30-35,

Hool, J. N. (1965). "A dynamic programming-probability approach to forest production control." **Soc. Am. For. Proc.** : 191-193,

Hool, J. N. (1966). "A dynamic programming-Markov chain approach to forest production control." **For. Sci. Monogr. 12**: 26,

Horne, R. R. and G. Robinson. (1988). Response Increment, the basis of a stand growth model. For. Comm. NSW Report,

IEEE Inc. (1984). Standard Dictionary of Electrical and Electronic Terms. New York,

Innis, G. and Rexstad, E. (1983) Simulation model simplification techniques. In: Sands, P. (1988) Is small beautiful? Pros and cons of simple, appropriate models,

Jenkins, G. M. and D. G. Watts. (1968). Spectral Analysis and its applications. San Francisco, Carlifornia, Holden-Day, Inc.,

Jennings, L.S., Fisher, M.E., Teo, K.L., and Goh, C,J. (1990) Miser: Optimal Control software-Theory and user Manual. EMCOSS Pty. Ltd., W.A. 6020, Australia,

Johnson, R. C. (1988). Lectures on Adaptive Parameter Estimation. Prentice Hall advanced Reference Series. Englewood Cliffs, New Jersey, Prentice Hall,

Johnston, D. R., A. J. Grayson and R. T. Bradley. (1967). Forest Planning. London, Faber and Faber Limited,

Kailath, T. (1980). Linear systems. Englewood Cliffs, N.J., Prentice-Hall,

Kassier, H.W. (1991). Revised thinning policy for softwood sawlog production. Unpublished addendum to Forest Management Instruction No 1/3/1. Department of Water Affairs and Forestry, South Africa pp.18,

Katz, S. (1962). "A Discrete Version of Pontryagin's Maximum Principle." **J. Electronics and Control. 13**: p 179,

Kilkki, P. and U. Vaisanen. (1969). Determination of the optimal cutting policy for the forest stand by means of dynamic programming. Forestallia, Fennica, Acta,

Knoebel, B.R., Burkhart, H.E. and Beck, D.E. (1986) A growth and yield model for thinned stands of yellow poplar. **For. Sci., Monograph 27**,

Kreyszig, E. (1983). Advanced Engineering Mathematics. United States of America, John Wiley & Sons, Inc.,

Kuczera, G. (1985). Prediction of water yield reductions following a bushfire in ash - mixed species eucalypt forest. Water Supply Catchment Hydrology Research Report,

Kuo, T., B. C. Kuo and C. Kuo. (1962). Automatic control systems. Prentice-Hall electrical engineering series. Englewood Cliffs, N.J., Prentice Hall,

Larson, P.R. (1963) Stem Form Development of Forest Trees. **For. Sci. - Monograph 5**,

Landsberg, J. J., W. A. Thompson, R. E. McMurtie, S. Linder and L. Wolf. (1985). Development and use of a complex model of growth of radiata pine. In: Modelling Trees, Stands and Forests. Edited by J. W. Leech, R. E. McMurtie, P. W. West, R. D. Spencer and B. M. Spencer. University of Melbourne. School of Forestry, University of Melbourne. Bulletin No. 5. 256-264,

Lau, J.A. Vandenberg, W.G. Willig R.U. (1994). Visual and spatial techniques in multiple-use forest planning. Resource Technology 94. 154-77,

Lee, E. S. (1963). "Optimization by Pontryagin's Maximum Principle on the Analog Computer." **JACC.** : pp 524-531,

Leigh, T.R. (1992) Control Theory, a guided tour. Peter Peregrinus Ltd., London, United Kindom.

Lembersky, M.R. and Chi, U.H. (1984) "Decision Simulators" Speed implementation and improve operations. **INTERFACES 14 no.4:1-15**. An International Journal of the Institute of Management Sciences and the Operatrions Research Society of America.

Lewin, R. (1992). Making maths make money. **New Scientist**. No. 1816. 31-34. 11/04/92,

Lipschutz, S. (1987). Theory and Problems of Linear Algebra. Schaum's Outline Series. Singapore, McGraw-Hill Inc.,

Little, S.N. (1976) The Weibull Function as a Diameter Distribution Model for Mixed Stands of Douglas-Fir and Western Hemlock. Master of Science Thesis, Pennsylvania State University,

Ljung, L. (1987). System Identification: Theory for the User. Englewood Cliffs, New Jersey 07632, Prentice-Hall, Inc.,

Ljung, L. (1991). System Identification Toolbox for use with MATLAB- User's Guide. Natick, Mass. 01760, The MathWorks, Inc.,

Ljung, L. and Soderstrom, T. (1983). Theory and Practice of Recursive Identification. Cambridge, Mass., MIT Press,

Luenberger, D. G. (1979). Introduction to dynamic systems: theory, models, and applications. New York, Wiley,

Maine, R. E. and K. W. Iliff. (1985). Identification of dynamic systems : theory and formulation. NASA reference publication ; no.1138. Washington : National Aeronautics and Space Administration, Scientific and Technical Information Branch,

Martin, G. L. and A. R. Ek. (1981). "A dynamic programming analysis of silvicultural alternatives for red pine plantations in Wisconsin." **Can J Forest Res. 11**: pp 370-379,

MathWorks Inc. (1992) MatLab: High performance Numeric Computations and Visualisation Software, Reference Guide. Cochituate Place, Natik, Mass. 01760,

McMullan, M. J. (1978). Prediction of results from possible alternative management regimes. Computer Growth Simulation. Tumut. Forestry Commission of NSW,

Meisel, W.S. and Collins, D.C. (1973). Repro-modelling: an approach to efficient model utilisation and interpretation. In: Sands, P. (1988) Is small beautiful? Pros andcons of simple, appropriate models,

Menzies, M.I., Klomp, B.K., Holden, D.G. and Hong, S.O. (1983). The effects of initial spacing on growth and crop selection of radiasta pine seedlings and cuttings. Efficiency of Stand Establishment Operations; IUFRO, Ed., Menzies, M.I., Parrott, G.E. and Whitehouse, L.J., FRI Bulletin 156. pp 152-165,

Munro, D. (1973). Forest Growth Models - a Prognosis. In: Growth models for Tree and Stand Simulation. The Swedish University of Agricultural Sciences. Department of Forest Yield Research. Research paper no. 30, pp 7-21,

Nahorski, Z., H. F. Ravn and R. V. V. Vidal. (1983). Optimization of Discrete Time Systems: The Upper Boundary Approach. Lecture Notes in Control and Information Sciences. Springer-Verlag, Beltz Offsetdruck,

Naslund, B. (1969). "Optimal rotation and thinning." **For. Sci. 15**: 446-451,

Nemhauser, L. G. (1966). Introduction to dynamic programming. New York, NY, John Wiley & Sons, Inc.,

O'Connor, A.D.J. (1935) Forest research with special reference to planting distances and thinning. British Empire Forestry Conference, South Africa,

Oliver, C.D. and Larson, B.C. (1990) Forest Stand Dynamics. Biological Resource Management Series. McGraw-Hills Inc. United States of America,

Oppenheim, A. V., A. S. Willsky and I. T. Young. (1983). Signals and systems. Englewood Cliffs, N.J., Prentice Hall,

Paredes-Veloso, G.L. and J. D. Brodie. (1985). Efficient computation considerations in optimization of stand-level management regimes. P. E. Dress and R. C. Field. The 1985 Symposium on System Analysis in Forest Resources. Georgia Centre for Continuing Education, Athens, Georgia. Georgia Centre for Continuing Education. 292-302,

Pearse, P.H. (1990). Introduction to Forestry Economics. The University of British Columbia Press. Vancouver, Canada,

Pelkki, M.H. (1994). Exploring the effects of aggregation in dynamic programming. In: Management Systems for a Global Economy with Global Resource Concerns, Edited by Sessions, J. and Brodie, J.D., Pacific Grove, Carlifornia.

Philips, C. L. and R. D. Harbor. (1991). Feedback Control Systems. Englewood Cliff, Prentice-Hall, Inc.,

Pienaar, P.V. (1979). An approximation of basal area growth after thinning based on growth in unthinned plantations. **For. Sci. 25(2):** pp 223-232,

Pienaar, L. V. and W. M. Harrison. (1994). Methodology to Evaluate Plantation Management Alternatives. Forestek - CSIR Report,

Pindyck, R. S. and D. L. Rubinfeld. (1981). Econometric models and economic forecasts. New York, Mcgraw-Hill,

Pnevmaticos, S.M. and Mann, S.H. (1972) "Dynamic programming in tree bucking," **Forest Products Journal, 22 no.2: 26-30**,

Pontryagin, L. S. (1959a). Basic Problems of Automatic Regulation and Control. Nauk, USSR, Izd-vo Akad (in Russian), (in Nahorski et al., 1983),

Pontryagin, L. S. (1959b). "Optimal Regulation Processes." **Uspekhi Matem. Nauk. 14**(No. 1): (in Russian), English translation in Am. Math. Soc. Trans., Ser.2, Vol. 18, 321-339, 1961,

Pontryagin, L. S. (1959c). Some Mathematical Problems Arising in Connection with the Theory of Optimum Automatic Control Systems. Academy of Sciences USSR on Scientific Problems of Automating Industry. USSR. (in Russian), (in Nahorski et al., 1983),

Pontryagin, L. S., V. S. Boltyanskii, R. V. Gamkrelidze and E. F. Mishchenko. (1962). The Mathematical Theory of Optimal Process. New York, Wiley - Interscience,

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992). Numerical Recipes in C: The Art of Scientific Computing. 2nd Edn., Cambridge University Press.

Propoi, A. I. (1964). "On a problem of Optimal Discrete Control." **Doklady Akad. Nauk SSSR. 159**(6): (in Nahorski et al., 1983),

Raven, P.H., Evert, R.F. and Curtis, H. (1982). Biology of Plants. Worth Publishers, Inc., New York 10016,

Rimmington, G. M. (1985). Simulation programming languages and modelling in forestry. In: Modelling Trees, Stands and Forests. Edited by J. W. Leech, R. E. McMurtie, P. W. West, R. D. Spencer and B. M. Spencer. University of Melbourne. School of Forestry, University of Melbourne. Bulletin No.5. 34-60,

Risvand, J. (1969). Economic analysis of cutting programs applying dynamic programming. Readings in Forest Economy. Oslo, Norway, Universitetsforlaget,

Rosen, R. (1971). A Survey of Dynamical Descriptions of Systems Activity. In: Unity Through Diversity,

Rong-wei, L. (1992). Improved Diameter Growth Models for Mountain Ash Plantation. PhD Thesis, University of Melbourne, Victoria, Australia,

Rozonoer, L. I. (1959). "The Maximum Principle of L.S. Pontryagin in the Theory of Optimal Systems." 10-12,

Sands, P. (1988). Is small beautiful? Pros and cons of simple, appropriate models. In: Modelling Trees, Stands and Forests. Edited by Leech, J.W., R.E. McMurtie, P.W. West, R.D. Spencer. Bulletin No. 5, School of Forestry, University of Melbourne,

Schlaegel, B.E. (1971). Growth and yield of quaking aspen in north central Minnesota. U.S.D.A. **For. Serv., Res. Paper NC-58**,

Schmidt, W. C. (1978). Some biological and physical responses to forest stand density. Eighth World Forestry Congress. Jarkarta, Indonesia. P.T. Gramedia, Indonesia. FQL/ 25 - 2. pp 12,

Schreuder, G. F. (1968). Optimal forest investment decisions through dynamic programming. Yale Univ School of Forestry,

Schreuder, G. F. (1971). "The simultaneous determination of optimal thinning schedule and rotation for even-aged forest." **For. Sci. 17**: 333-339,

Schumacher, F.X. (1939). A new growth curve and its applicationto timber-yield studies. **J. Forest** 37: 819-820,

Shannon, R. E. (1975). Systems Simulation: The Art and Science. Englewood Cliffs, N.J., Prentice Hall,

Shepherd, K.R. (1986) Plantation Forestry. Martinus Nijhoff Publishers, Dordrecht, The Netherlands,

Sloboda, B. (1984). Possibilities for mathematically predicting timber production in commercial forests. Bulletin of Nagoya University Forest 7: 261-279,

Society of American Foresters, (1971). Terminology, practice and products. Society of American Foresters. Washington D.C.,

Stoate, T.N. (1942). "Taper in *Pinus pinaster*". **Aust. For. 6**: 59-66,

Sullivan, A.D. and Clutter, J.L. (1972). A simultaneous growth and yield model for loblolly pine. **For. Sci. 18**: 76-86,

Suzuki, T. (1971). Forest transition as a stochastic process. In: Contribution to increment research. Forrstliche Bundesversuchsanstalt, Austria. pp 69-83,

Tanaka, K. (1988). Prediction of diameter distribution in an even-aged pure stand, In: Proceedings of Japan and New Zealand Symposium on Forestry Management Planning. Forest Research Institute, Rotorua, pp 111-129,

Turner, B.J. (1995). The forest planners' fundamental questions: where and when?. In: Applicaction of New Technologies in Forestry. Edited by L. Bren and C. Greenwood. Ballarat, Victoria. Institute of Foresters of Australia. vol (1): 239-245,

Valsta, L.T. (1994). Silvicultural guidelines based on optimisation with stochastic price and growth. In: Management Systems for a Global Economy with Global Resource Concerns, Edited by Sessions, J. and Brodie, J.D., Pacific Grove, Carlifornia.

Vandaele, W. (1975). Applied Time Series and Box-Jenkins Models. New York, N.Y. 10003, Academic Press, Inc.,

Van Laar, A. (1976) Thinning research in South Africa. In: Hamilton, G.J. (ed), Aspects of thinning. Forestry commission bulletin 55. H.M.S.O. London. pp.62-71,

von Gadow, K. (1983) A model of the Development of Unthinned *Pinus patula* Stands, **South African Forestry Journal 126:pp 39-47**,

Wagner, H. M. (1975). Principles of Operations Research - with applications to managerial decisions. Englewood Cliffs, NJ, Prentice-Hall,

Weisberg, S. (1985). Applied linear regression. Wiley series in probability and mathematical statistics. New York, Wiley,

Welch, S.M., Croft, R.H., and Michels, M.F. (1981) Validation of pest-management Models. In: Sands, P. (1988) Is small beautiful? Pros and cons of simple, appropriate models,

Wessels, N.O. (1977) Dunningsbeleid vir *P. patula, P. elliottii* end *P. taeda* vir saaghoutproduksie. Unpublished addendum to forest management instruction No. 58. Department of Forestry, South Africa,

West, P. W., C. L. Beadle and C. R. A. Turnbull. (1989). "Mechanistically based, allometric models to predict tree diameter and height in even-aged monoculture of *Eucalyptus regnans* F. Muell." **Canadian Journal of Forest Research. 19**: 270-273,

Willems, J. C. (1979). "System theoretic models for the analysis of Physical Systems." **Automatica. 10(2)**: 71-106,

Willems, J. C. (1986a). "From time series to linear system - Part I. Finite dimensional linear time invariant systems." **Automatica. 22(5)**: 561-580,

Willems, J. C. (1986b). "From time series to linear systems - Part II. Exact modelling." **Automatica. 22(6)**: 675-694.

APPENDIX I

Choosing a Model Class

Choosing a model structure is of crucial importance. The usual approaches towards providing algorithms for a model are:

(a) a collection of results from some experiment based on careful thought and reason;

(b) formulation of a theory from (a) that is analysed against all possible consequences to see if anything else is implied; and

(c) postulation of a set of equations containing as yet unspecified parameters.

The parameters in (c) are then fitted by means of the data. In the process of setting up such a procedure one has to face the conceptual difficulty that many models that one would like to use will be unable to adequately explain the data. The standard way out of this dilemma is to invoke the philosophy of statistics and employ models which contain, in addition to unspecified parameters, random elements (Willems, 1986b), for example, by assuming that the observed time series is a realisation of a stochastic process, or that it is the output of a system driven by a stochastic process.

Such an assumption then usually guarantees that every (finite) observed time series can occur with a certain probability and in this sense the data will not refute the model. The modelling question is then one of sampling: one has to deduce from the observations the probability measure governing the random system (Willems, 1986b).

Granted, there are many situations in which such a framework is indeed a suitable one. However, as a general philosophy, it has many fundamental drawbacks. The main short-coming is that the lack of fit between data and model is not in the first

place due to randomness or measurement noise but to the fact that one consciously uses a model whose structure is unable to capture the complexity of the phenomenon that one is observing. Sometimes it is only other thoughts or theories that are required to modify a function that will not fit. For example, Sir Isaac Newton formulated the Law of Gravitation between two bodies that they exert a force upon each other that varies inversely as the square of the distance between them, and varies directly as the product of their masses. From this law, Newton was able to demonstrate the elliptical motion of the planets around the sun. More recently, in the beginning of the twentieth century, it became apparent that the motion of the planet Mercury was not exactly right. Albert Einstein had to modify the Laws of Gravitation in accordance with his principles of relativity. The modification had very small effects but just the right amount to account for the slight discrepancy that was found in the movement of Mercury.

However, dynamical model structures have been demonstrated to adequately account for tree stand growth (Chikumbo et. al, 1992; Chikumbo and Mareels, 1993) and there seems no reason to venture into stochastic modelling or any other new theories/laws for modification. Note that the previous assertion hinges on the final application of the models and that is control. Thus concentration is on (c) and this appendix looks at one of the tools available for system identification and how to interpret the results from it, in order to extract the 'best' model. MatLab's system identification toolbox is demonstrated and its capabilities are illustrated by using data from a case study problem that comes with the computer package. All the MatLab commands are written in italics. The data has the basic input/output structure over some time period and in that sense is typical of the tree stand growth data that was analysed for this thesis.

I.1 Building Simple Models from Process Data

The case study in this section concerns data collected from a laboratory scale 'hairdryer' (p 440 in Ljung, 1987). Air is blown through a tube after being heated at the inlet to the tube. The input to the process is the voltage applied to a mesh of resistor wires that constitutes the heating device. The output of the process is the air temperature at the outlet measured in volts by a thermocouple sensor.

One thousand input/output data points were collected from the process as the input was changed in a random fashion between two levels with probability 0.2. The sampling interval is 80ms. The data is stored as the vectors *y2* (output) and *u2* (input) in some data file.

To build the model, the first 300 data points are used and for convenience, the input/output vectors are merged into a matrix:

z2 = [y2(1:300) u2(1:300)];

To look at the data,

idplot(z2)

is used which graphs the output on top and the inputs at the bottom.

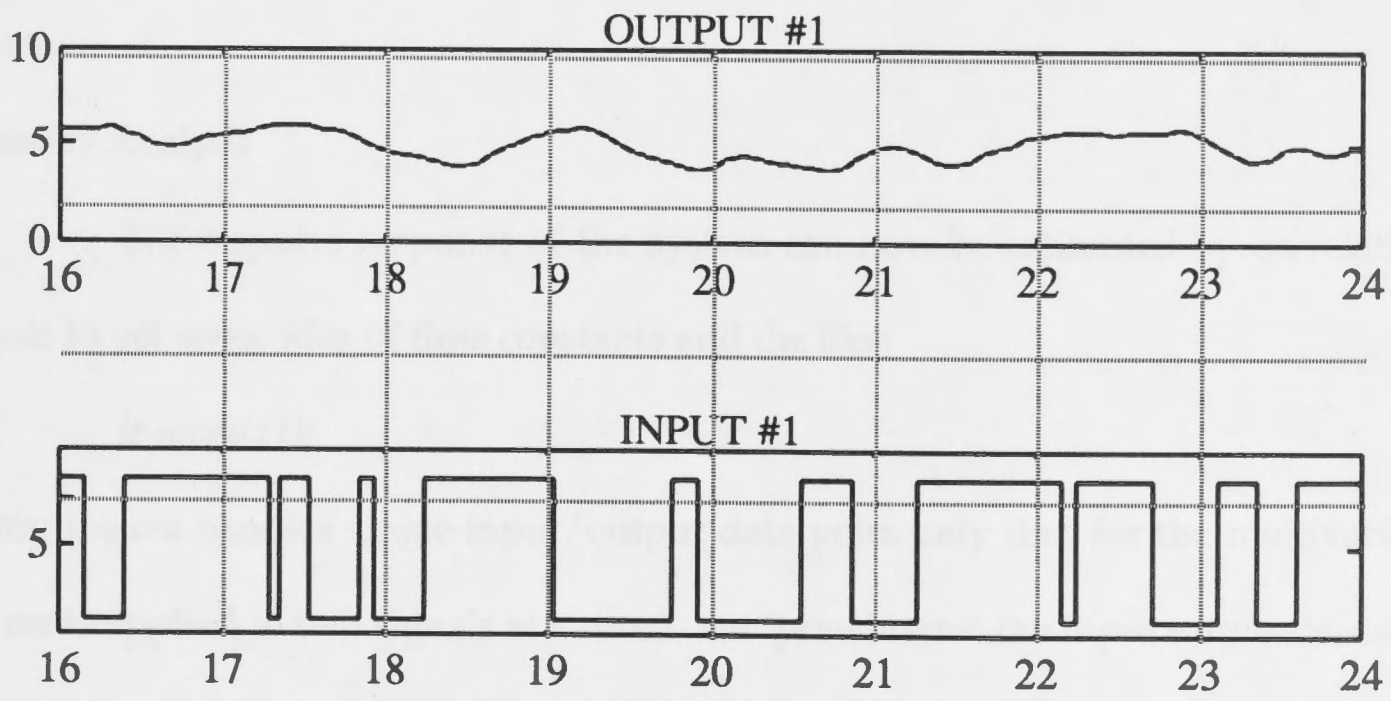


Figure I.1: Plots of the observed data from sample points in the range of 200-300 (MatLab output).

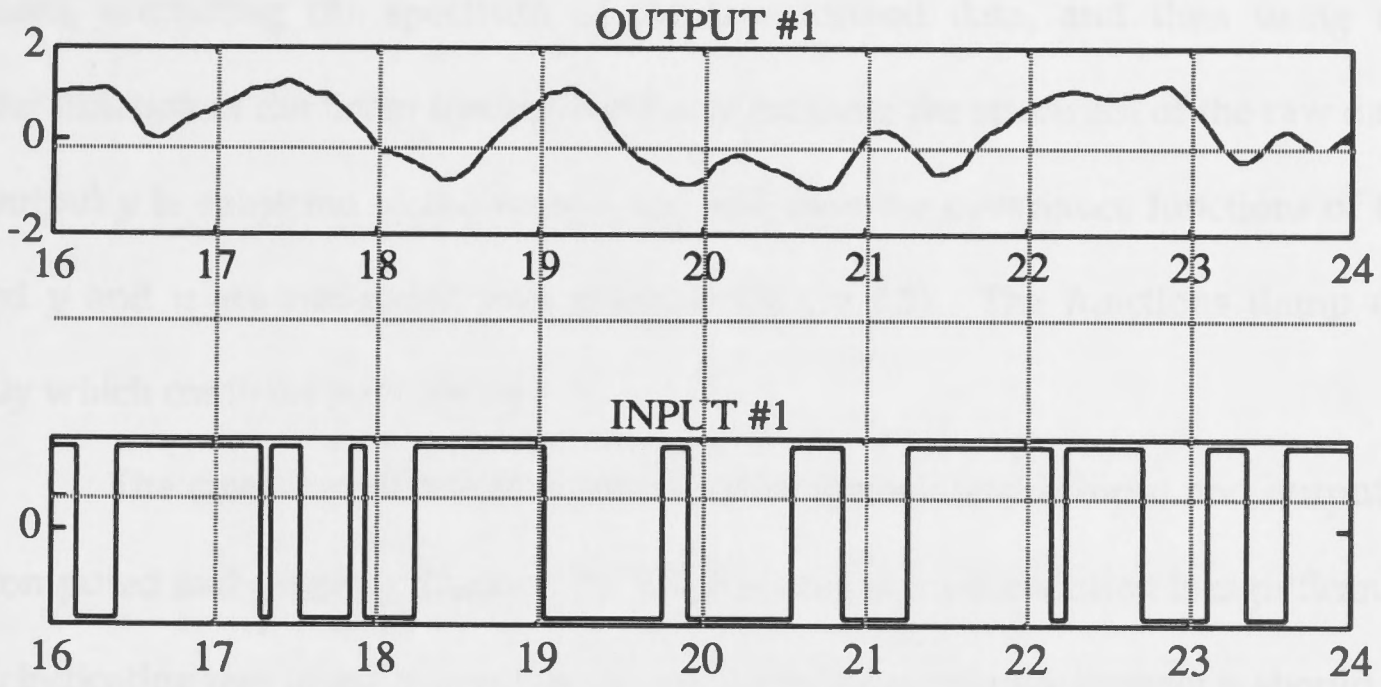


Figure I.2: Plot of Stationary data points in the range of 200-300 (MatLab output).

For a close-up, the values between sample numbers 200 and 300 have been selected:

idplot(z2, 200: 300, 0.08)

Figure I.1 shows the plot. The constant levels are removed (detrending) where the data takes a zero mean, i.e. the sample means are subtracted from each column:

z2 = dtrend(z2);

Note that the plots do not change (figure I.2) save the scales, because no trend exists in the data.

Correlation Analysis

The *i*mpulse *r*esponse of the system can now be estimated by correlation analysis to get some idea of time constants and the like:

ir = cra(z2);

The routine *cra* handles single-input/output data pairs only (but for the multivariate case, *cra* is applied to two signals at a time). *cra* 'prewhitens' the input sequence, i.e. it filters *u* through a filter chosen so that the result is as uncorrelated (white) as possible. Prewhitening consists of making a linear transformation of the raw data so as to achieve a smoother spectrum (the cosine transformation of the autocovariance

function), estimating the spectrum of the transformed data, and then using the transfer function of the linear transformation to estimate the spectrum of the raw data. The output y is subjected to the same filter, and then the covariance functions of the filtered y and u are computed and graphed (figure I.3). The functions damp out quickly which confirms stationarity.

The cross correlation function between (prewhitened) input and output is also computed and graphed (figure I.3). The zero lag cross correlation is significantly small, indicating that at lag 0, y and u are not perfectly correlated: current u should be a perfect predictor of y at approximately three periods ahead, i.e. for this data there is a delay (dead) time parameter, $nk = 3$ (section I.3). Positive values of the lag variable then corresponds to an influence from u to later values of y. In other words, significant correlation for negative lags is an indication of feedback from y to u in the data (Ljung, 1991).

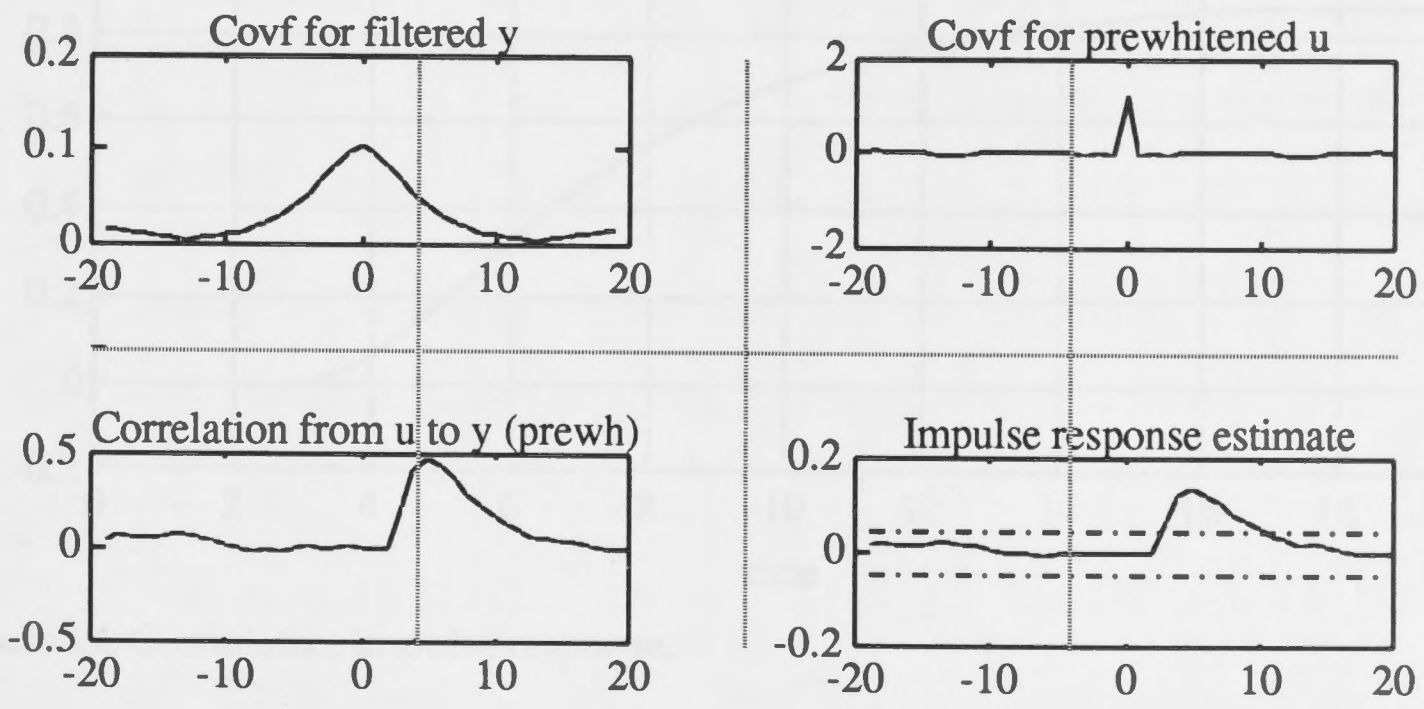


Figure I.3: Covariance functions of y and u, cross correlations of input/output and impulse response estimation (MatLab output).

A properly scaled version of this correlation function is also an estimate of the system's impulse response, *ir*. This is also graphed along with 99% confidence levels. The output argument, *ir*, is the impulse response estimate (figure I.3), so that its first entry corresponds to lag zero. Negative lags are excluded in *ir*. The corresponding step response can be obtained by integrating the impulse response:

stepr = cumsum(ir);

where, *cumsum(ir)* is the cumulative sum of the elements of *ir*. Figure I.4 shows the plot of *stepr* obtained by,

plot(stepr)

The graph in figure I.4 satisfies the restriction in equation (2.3). The unit step response can be used to characterise an LTI system since the impulse can be calculated from it.

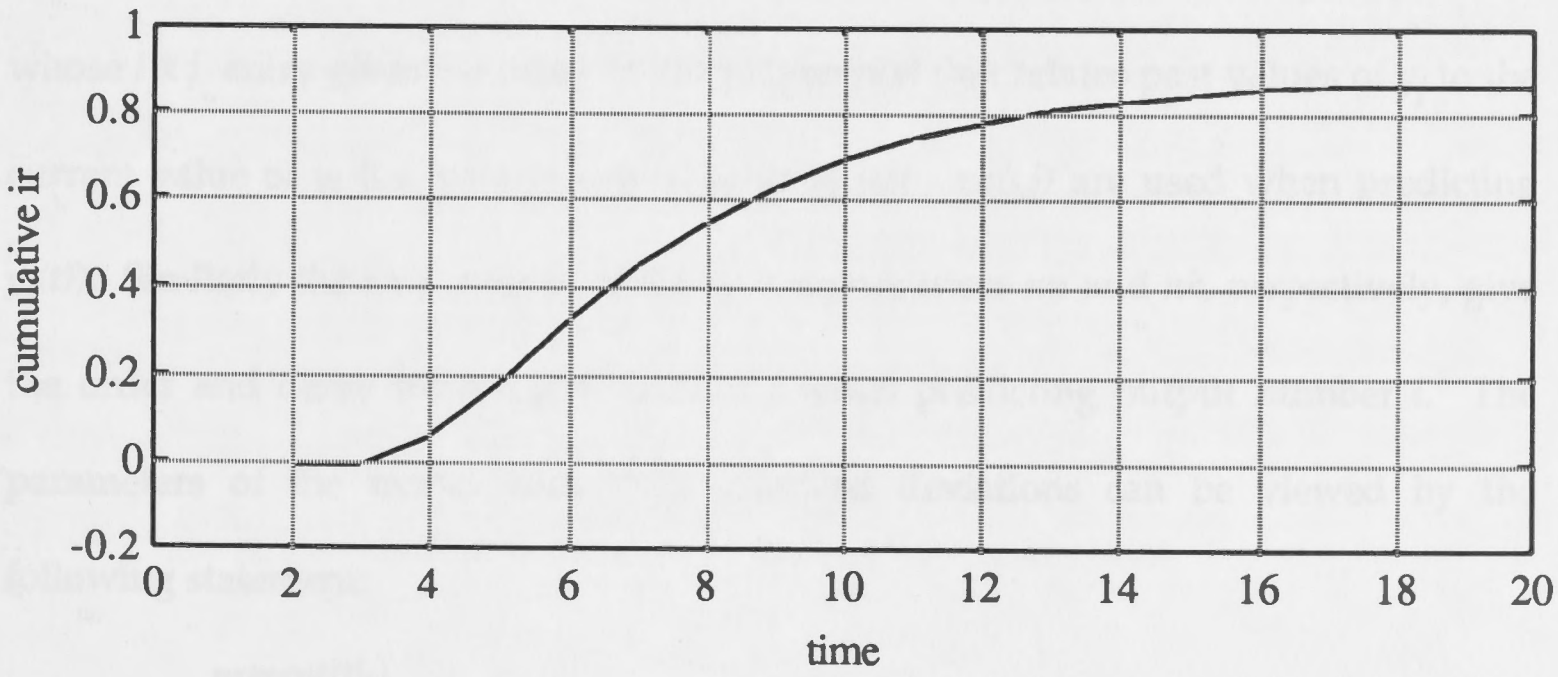


Figure I.4: Cumulative impulse response.

Now that the impulse response of the system has been obtained, the model can be constructed.

An ARX model can be computed with two poles, one zero and three delays:

th = arx(z2, [2 2 3]);

th = sett(th, 0.08); {This statement sets the sampling interval at 0.08 which would otherwise be defaulted to 1}

arx estimates the parameters a_i and b_i of the ARX model of the form $A(q)y(t) = B(q)u(t - nk) + e(t)$ or more explicitly as in equation (3.5a), using least squares estimation:

th = arx(z, [na nb nk])

z has been defined before as a matrix that consists of two columns, *y* and *u*. *na*, *nb* and *nk* are corresponding orders and delay that define the structure of the model. Note that *arx* is also applicable to arbitrary multivariable systems (Ljung, 1991). If there are *ny* outputs and *nu* inputs, the orders are defined accordingly: *na* is an *ny* x *ny* matrix whose i x j entry gives the order of the polynomial that relates past values of y_j to the current value of y_i (i.e. past values of y_j up to $y_j(t - na(i,j)$ are used when predicting $y_i(t)$). Similarly the i x j entries of the *ny* x *nu* matrices *nu* and *nk*, respectively, give the order and delay from input number j when predicting output number i. The parameters of the model with their standard deviations can be viewed by the following statement:

present(th)

The polynomial coefficients (top row) and their standard deviations (bottom row) are shown below:

B =

0	*0*	*0*	*0.0666*	*0.0445*
0	*0*	*0*	*0.0021*	*0.0033*

A =

1.0000	*-1.2737*	*0.3935*
0	*0.0208*	*0.0190*

A is thus interpreted as $A(q) = 1 - 1.2737q^{-1} + 0.3935q^{-2}$ and the standard deviation of '1' is zero (since it is not estimated) and the standard deviation of a^1 is 0.0208. The leading zeros in the *B* polynomial indicate the delay. Thus the ARX model is represented as:

$$y(t) - 1.2737y(t - 1) + 0.3935y(t - 2) = 0.0666u(t - 3) + 0.0445u(t - 4) \qquad \text{(I.1)}$$

One way of validating such a model would be to simulate it and compare the model output and the measured output. A portion of the original data that was not used to build the model is selected, viz from sample 800 to 900:

```
u = dtrend(u2(800:900)); y = dtrend(y2(800:900));

yh = idsim(u, th);
```

idsim returns *yh* containing the simulated output and for a graphical comparison, a plot of *yh* and *y* is plotted (figure I.5):

```
plot([yh, y])
```

The agreement is quite good.

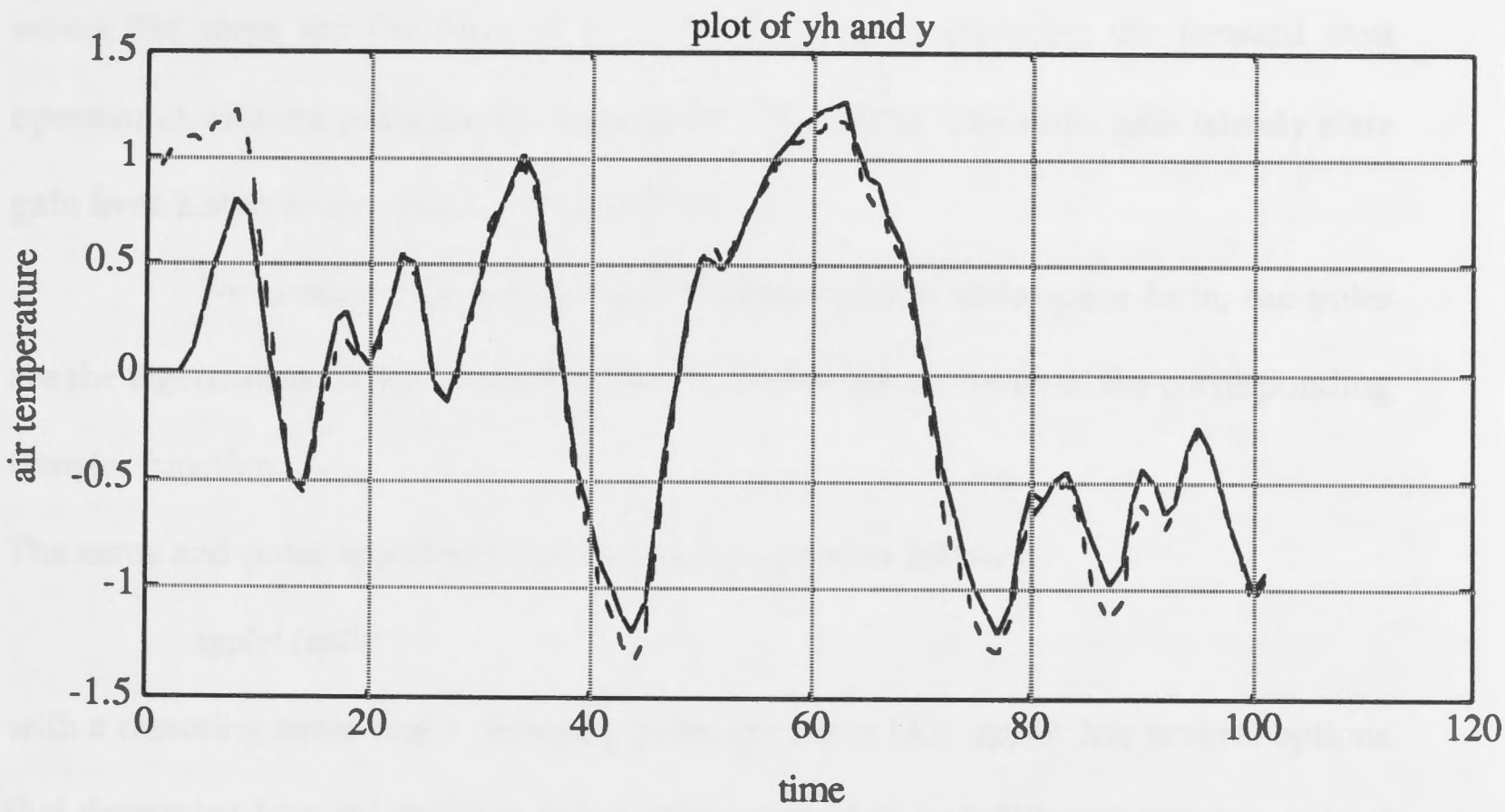


Figure I.5: Comparison of the model output and measured data

Determining uncertainty in a model

The uncertainty in internal representations of a model is manifested in the covariance matrix of the estimated parameters (by specifying *present (th)*) and also in the standard deviations of the pole and zero locations. The poles and zeros of the model are calculated as follows:

zpth = th2zp(th);

For any model described in the theta format by *th*, the poles and zeros and static gains, along with their standard deviations are computed. The poles and zeros are stored in a coded form in the matrix *zepo*, while the static gains are returned in *k*. The information can also be retrieved with *getzp*. The procedure handles both models in continuous and discrete times. For the general discrete-time multi-input/single output model takes the form:

$$A(q)y(t) = \frac{B_1(q)}{F_1(q)}\, u_1(t - nk) + \ldots + \frac{B_{nu}(q)}{F_{nu}(q)}\, u_{nu}(t - nk_{nu}) + \frac{C(q)}{D(q)}\, e(t) \qquad \text{(I.2)}$$

where the zeros are the roots of $z^{nb + nk} B(z)$ {with z replacing the forward shift operator q}, and the poles are the roots of $z^{na + nf} A(z)F(z)$. The static gain (steady state gain from a step in input) is $k = B(1)/A(1)F(1)$.

For models that are internally represented in state-space form, the poles are the eigenvalues of the system matrix. The zeros are the zeros of the corresponding transfer function.

The zeros and poles specified by *zepo* can be graphed as follows:

zpplot (zpth)

with *o* denoting zeros and *x* denoting poles (Endnote IA). *zpplot* has several options that determine how information about different models and different input or output is depicted. It also keeps track of whether the underlying model is in discrete or continuous time and draws the unit circle or complex plane accordingly. Poles and zeros at infinity and at the origin are ignored. Large uncertainties in these representations are caused by excessively high model orders, inadequate perturbation, or bad response-to-noise ratios.

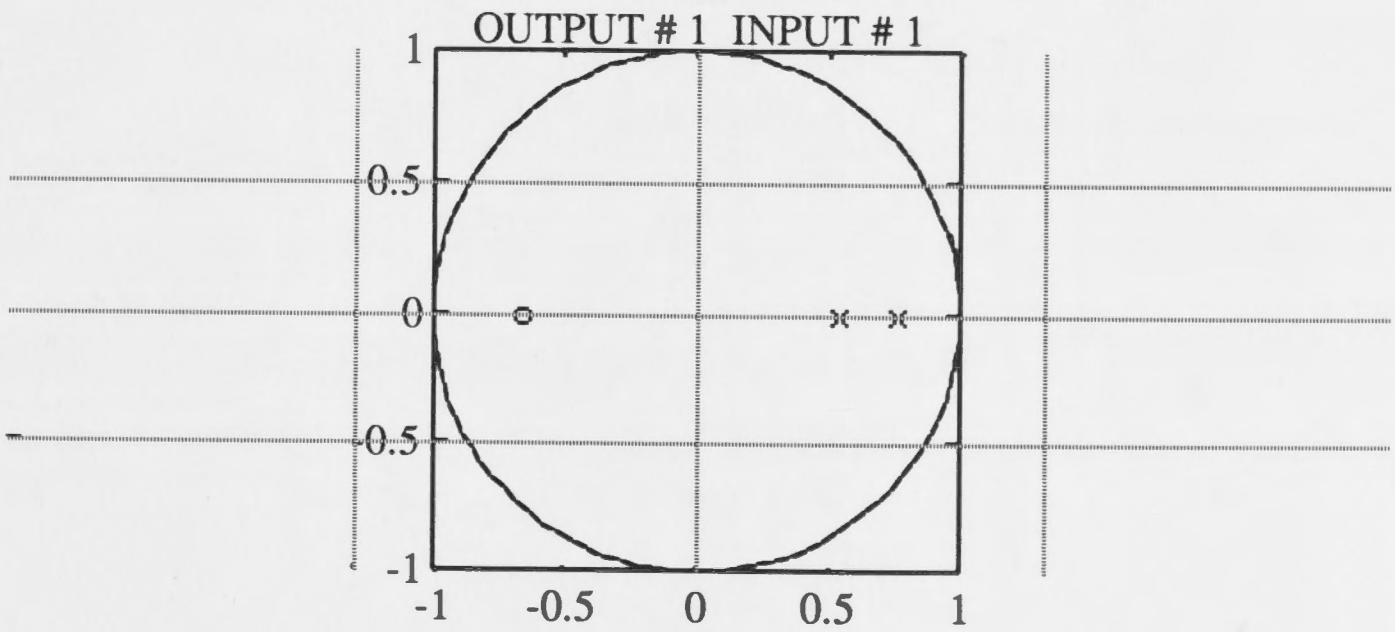


Figure I.6: Zero-pole plot (MatLab output)

Variability of a model with the same input sequence

The estimated model is always uncertain, due to disturbances in the observed data, and due to the lack of a correct model structure. The variability of the model, that is due to the random disturbances in the output, is estimated by most of the estimation procedures and can be displayed in a number of ways. This variability answers the question of how different the model can be if the identification procedure is repeated, using the same model structure, but with a different data set that uses the same input sequence (Ljung, 1991). It does not account for systematic errors due to an inadequate choice of model structure.

A MatLab function defaults to 10 different step responses corresponding to models drawn from the distribution of the true system, i.e. random models created according to the covariance information given in *th*:

idsimsd(step, th)

idsimsd simulates the theta format system with uncertainty. The responses of each of these models to the input step are computed and graphed in the same diagram (figure I.7). Figure I.7 seems to guarantee that the 'true system' lies in the confidence interval.

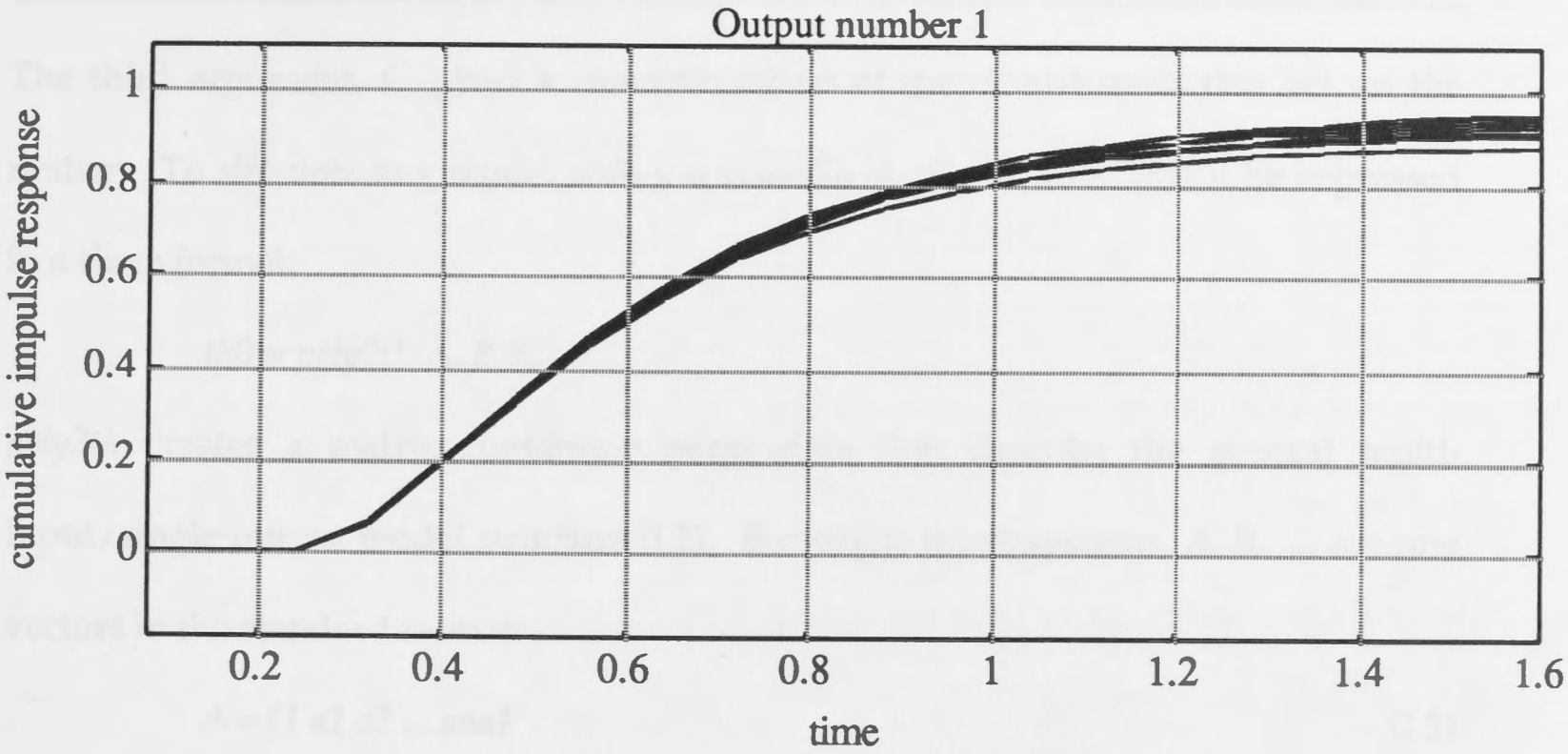


Figure I.7: Random models

The uncertainty of these responses concerns the external, input/output properties of the model. It reflects the effects of inadequate excitation and the presence of disturbances.

I.2 Comparison between different identification methods

The system identification toolbox contains several methods for parametric model estimation. They all share the same command structure

th = function(z, ths);

present (th)

To compare the different model structures, a time series data set will be generated by specifying the numerator and denominator coefficients of a transfer function model of a system:

B = [0 1 0.5];

A = [1 -1.5 0.7];

C = [1 -1 0.2];

The third argument, *C*, gives a characterisation of the disturbances that act on the system. To simulate any model with *y = idsim([u e], th)*, requires that it be expressed in a theta format:

th0 = poly2th(A, B, C);

poly2th creates a matrix containing parameters that describe the general multi-input/single-output model structure (I.2). For single input systems, *A*, *B*, ..., are row vectors in the standard format:

A = [1 a1 a2 ... ana] (I.3)

All the polynomials start with a *1*, with the exception of *B* which contains a leading zero to indicate the delays. Hence the system above has one delay (*nk = 1*).

The following string of commands generates an input signal *u*, a disturbance signal *e*, and simulates the response of the model to these inputs:

```
rand('normal')                 {switches to a normal distribution with
                                mean 0.0 and variance 1.0}
u = sign(rand(350, 1));        {For each element (n) the signum
                                function (sign) returns 1 for n > 0, 0 if
                                n = 0 and -1 if n < 0}
e = rand(350, 1);              {generates random numbers of a 350-
                                by-1 matrix}
y = idsim([u e], th0);
z = [y u]
```

Second order ARX model

The impulse response of this generated data is estimated by correlation analysis that suggests a second order model. An ARX model with two poles, one zero and a single delay is estimated using the least squares method:

```
a2 = arx(z, [2 2 1]);

present (a2)
```

The results are as follows:

Loss fcn: 1.912 *Akaike's FPE: 1.956* *Sampling interval 1*

B =

0	*0.9539*	*0.6224*
0	*0.0744*	*0.0831*

A =

1.0000	*-1.2159*	*0.4367*
0	*0.0394*	*0.0384*

The *loss function* (*Loss fcn*) is the best available estimate of the innovations (i.e. the sequence $\tilde{y}(t) = y(t) - \hat{y}(t)$) covariance and it can be used to compare different model structures. Care should be taken when comparing model structures that have very different noise models (Ljung, 1991). For most models the estimated covariance matrix of the innovations is obtained by forming the corresponding sample mean of the prediction errors. This is computed (using *pe*) from the model and the estimation data (Ljung, 1991).

Akaike's Final Prediction-error (FPE) criterion reflects the prediction-error variance that one will obtain, on average, when a model is applied as a predictor to other data sets than those used for the identification (Akaike, 1969). It is given as follows:

$$\mathrm{FPE} = \frac{1 + \mathrm{n}/\mathrm{N}}{1 - \mathrm{n}/\mathrm{N}} \mathrm{V} \tag{I.4}$$

where n is the total number of estimated parameters and N is the length of the data record. V is the loss function for the model structure in question.

Instrumental variable method

Another method called *instrumental variable method* (IV4) can be used to find a model with two poles, one zero and a single delay on the input:

```
i2 = iv4(z, [2 2 1]);

present (i2)
```

Suppose that a model description corresponds to equation (3.11). The idea is to derive instruments similar to equation (3.11) that are not influenced by $e(t)$. This leads to:

$$\zeta(t) = K(q)[-x(t - 1) - \ldots -x(t - n_a)\, u(t -1) + \ldots + n(t - n_b)] \qquad \text{(I.5)}$$

where K is a linear filter and $x(t)$ is obtained from the input $N(q)x(t) = M(q)u(t)$. This gives the instrumental variable method, and $x(t)$ are called the instruments (Ljung, 1987). The following results are obtained:

Loss fcn: 1.104 *Akaike's FPE: 1.13* *Sampling interval 1*

B =

0	*0.9937*	*0.4216*
0	*0.0558*	*0.0660*

A =

1	*-1.5140*	*0.7056*
0	*0.0159*	*0.0145*

It is difficult to compare the loss function in an ARX model that is estimated by using *arx* and one estimated by using *iv4*. This is because for ARX models computed using *iv4*, the covariance matrix of the innovations is estimated using the provisional noise model that is used to form the optimal instruments (Ljung, 1987).

The residuals (prediction errors) for the model obtained by IV4 can be calculated as follows:

e = resid(z, i2);

The autocorrelation function of *e* and the cross correlation between *e* and the input(s) *u* are computed and displayed (figure I.8). The 99% confidence intervals for these values are also computed and displayed as dotted curves. The rule is that if the correlation functions go significantly outside these confidence intervals, do not accept the corresponding model as a good description of the system (Ljung, 1991). Methods such as the IV, that focus on the dynamics of *G* and less on the noise properties of *H* always ensure that the correlation between *e* and *u* for negative lags, or current *e(t)* do not affect the future *u(t)*, i.e. there is no feedback of output. Figure I.8 shows a good model that has no feedback present in the input/output data. The computation of these values is done assuming *e* to be white and independent of *u* for the model to correctly describe the system (Ljung, 1991). The functions are displayed up to lag 25 by default.

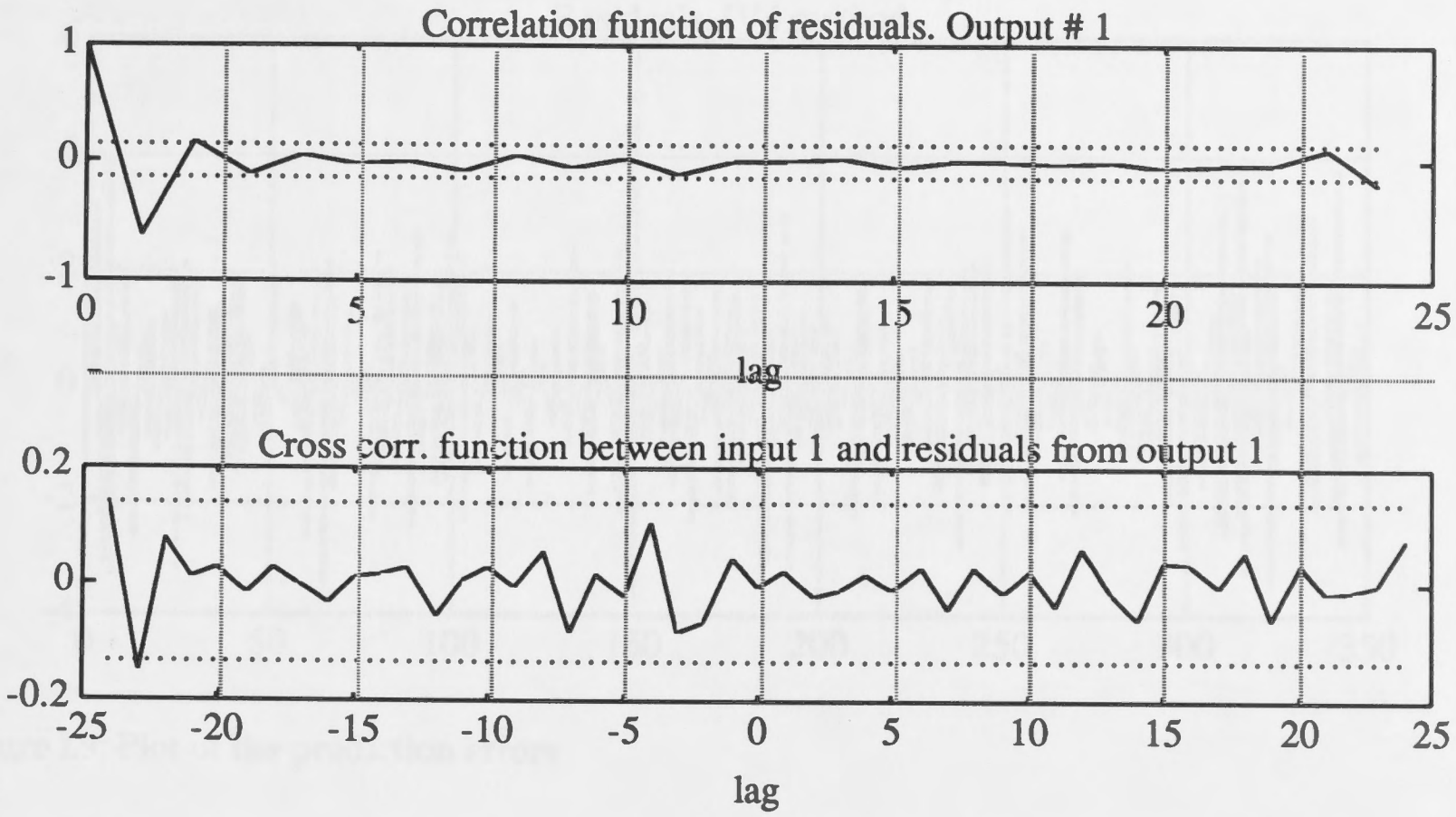


Figure I.8: Autocorrelation function of *e* and cross correlation between *e* and *u* from model obtained by IV4 (MatLab output).

The residuals plot is invoked as follows (figure I.9):

plot (e)

for a simple visual inspection of irregularities and outliers. It is always good practice to examine the residuals for unusually large values and to go back and critically evaluate the original data responsible for the large values. Figure I.9, however, does not show any unusually large disturbances. If the raw data were obviously in error, the outliers would be removed, and the estimation procedure repeated.

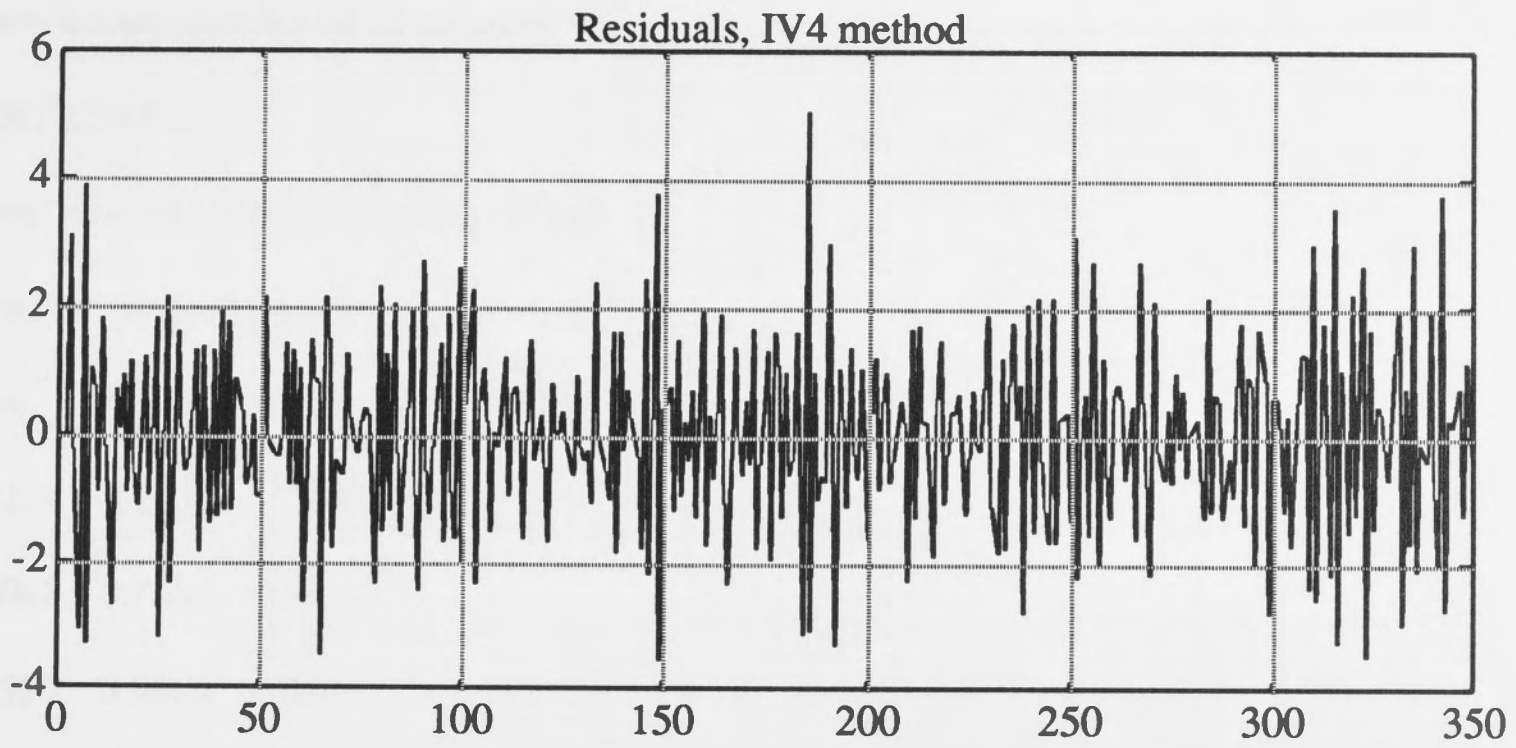


Figure I.9: Plot of the prediction errors

Second order ARMAX model

Another model structure that has a better noise prediction than the ARX function, is the ARMAX model, and it is also investigated. A second order ARMAX model is computed as follows:

am2 = armax(z, [2 2 2 1]);

The parameters of the ARMAX model structure $A(q)y(t) = B(q)u(t - nk) + C(q)e(t)$ are estimated using a prediction error method. Matrix *z* contains the input/output data structure, and *na, nb* and *nc* are the orders of the ARMAX model, and *nk* is the delay.

A robustified quadratic prediction error criterion is minimised by using an iterative Gauss-Newton (gn) algorithm (Ljung, 1987). Information about the minimisation is displayed on the computer screen during the computation. Current and previous parameter estimates (in column vector form, listing parameters in alphabetical order) as well as the values of the criterion function are given. The gn vector and its norm are also displayed. The number of iterations the search vector has been bisected is also shown.

The results are displayed as follows:

ITERATION # 3

Current loss: 0.9532 Previous loss: 0.9533

Current th prev. th gn-dir

theta =

-1.5174	*-1.5177*	*0.0003*
0.7087	*0.7089*	*-0.0002*
0.9960	*0.9965*	*-0.0005*
0.4201	*0.4177*	*0.0024*
-1.0523	*-1.0509*	*-0.0014*
0.1986	*0.1963*	*0.0023*

Norm of gn-vector: 0.003659

Figures I.10 shows the correlation and the cross correlation functions of the residuals for the ARMAX model and a good model is indicated.

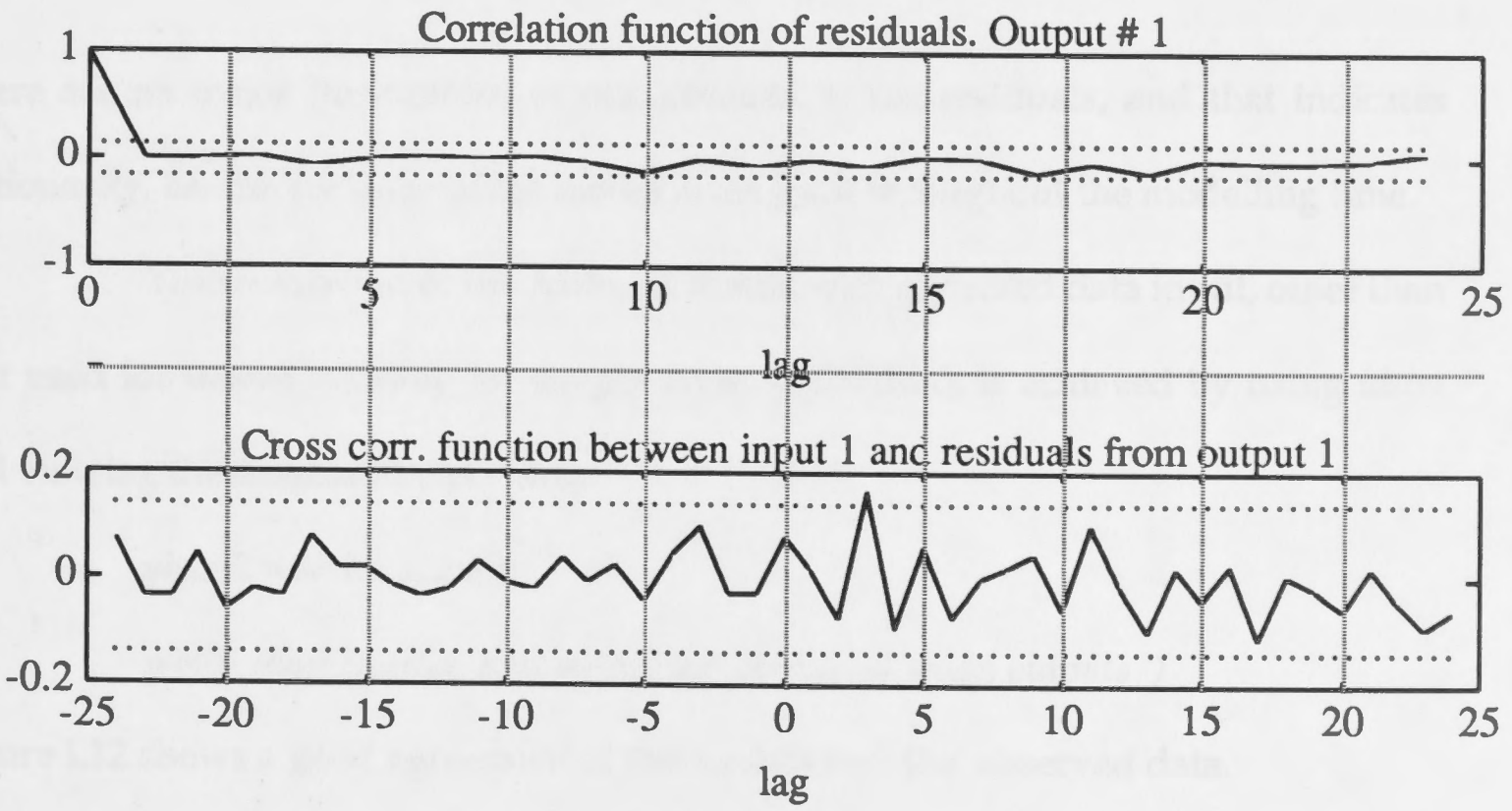


Figure I.10: Autocorrelation function of *e* and cross correlation between *e* and *u* from an ARMAX model (MatLab output).

The residuals of the ARMAX model are obtained by invoking the following statement:

e1 = resid(z, am2);

Figure I.11 shows the plot of the residuals of the ARMAX model:

plot(e1), title('ARMAX model residuals')

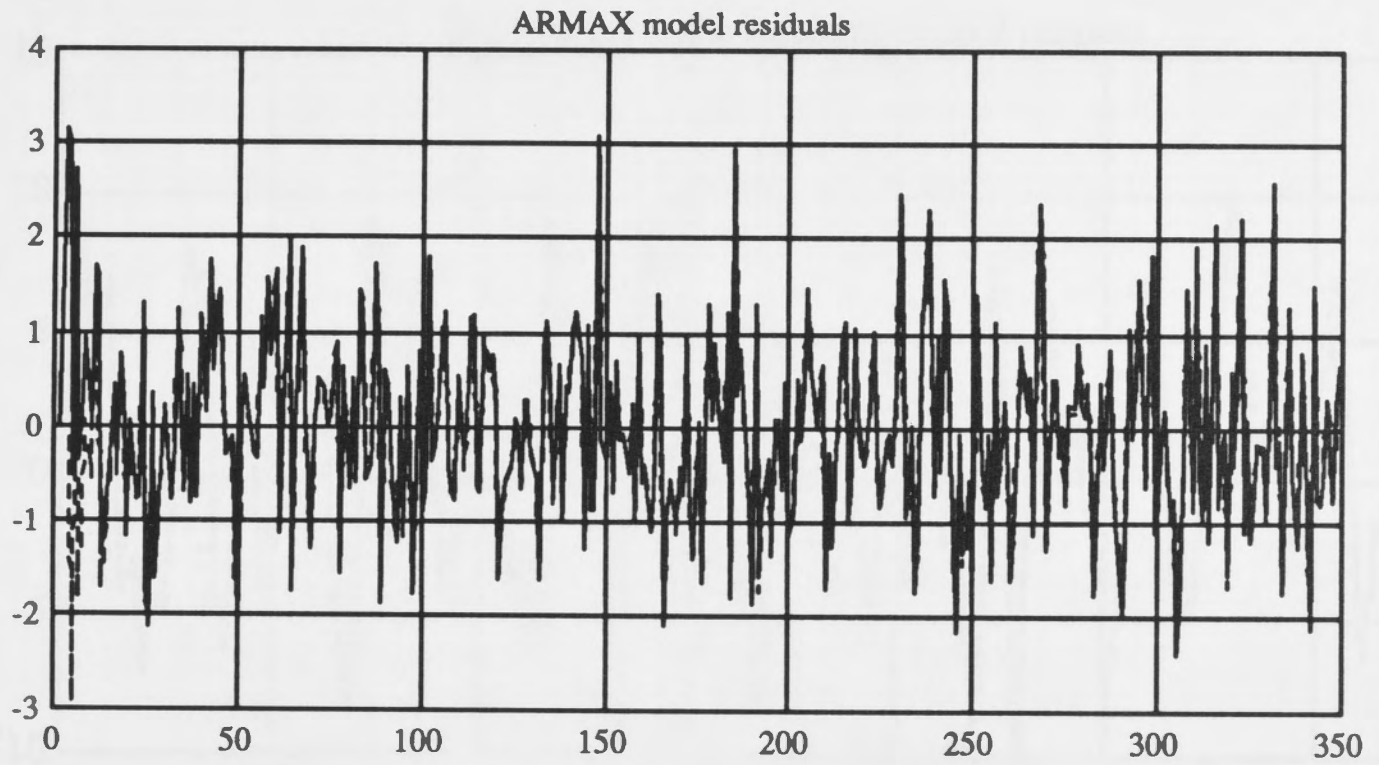


Figure I.11: Residuals from *armax* functions

There are no major fluctuations or disturbances in the residuals, and that indicates stationarity, i.e. the variance of the model is constant throughout the modelling time.

The simulation of the ARMAX model with observed data input, other than that used for model building (or simply cross validation), is achieved by using *idsim* and viewing the diagram by plotting:

yham2 = idsim(u, am2);

plot(y yham2); title('Real output and simulated model outputs ')

Figure I.12 shows a good agreement of the models and the observed data.

To check for stability of the model, the zeros and poles of the ARMAX model are computed as follows:

zpam2 = th2zp(am2);

The poles and zeros from the ARMAX model are plotted as follows (figure I.13):

zpplot(zpam2)

The ARMAX model is stable. Poles on the outside of the unit circle imply instability of the model, although the fit to the model might be statistically acceptable.

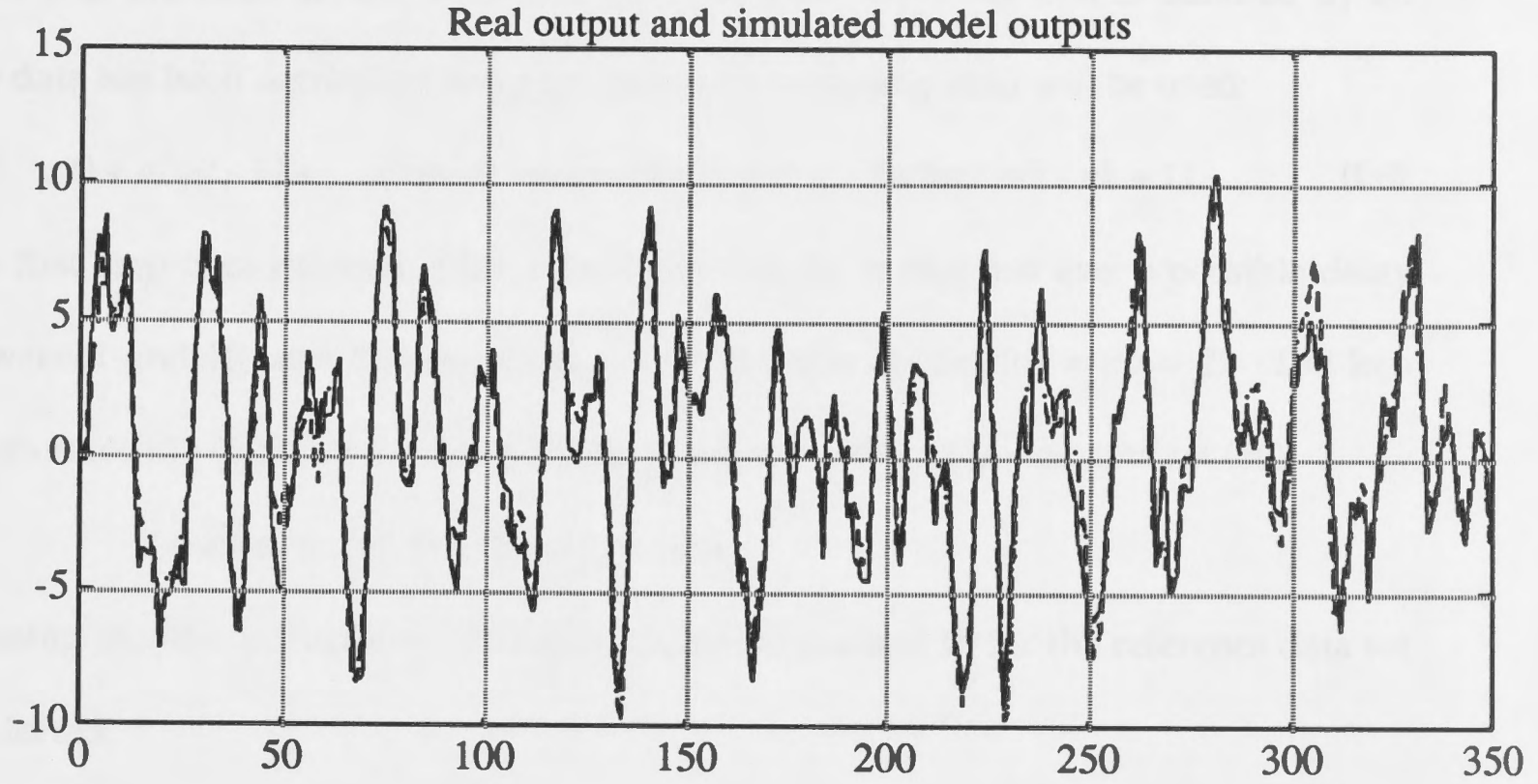


Figure I.12: Comparison of the ARMAX model prediction and measured data

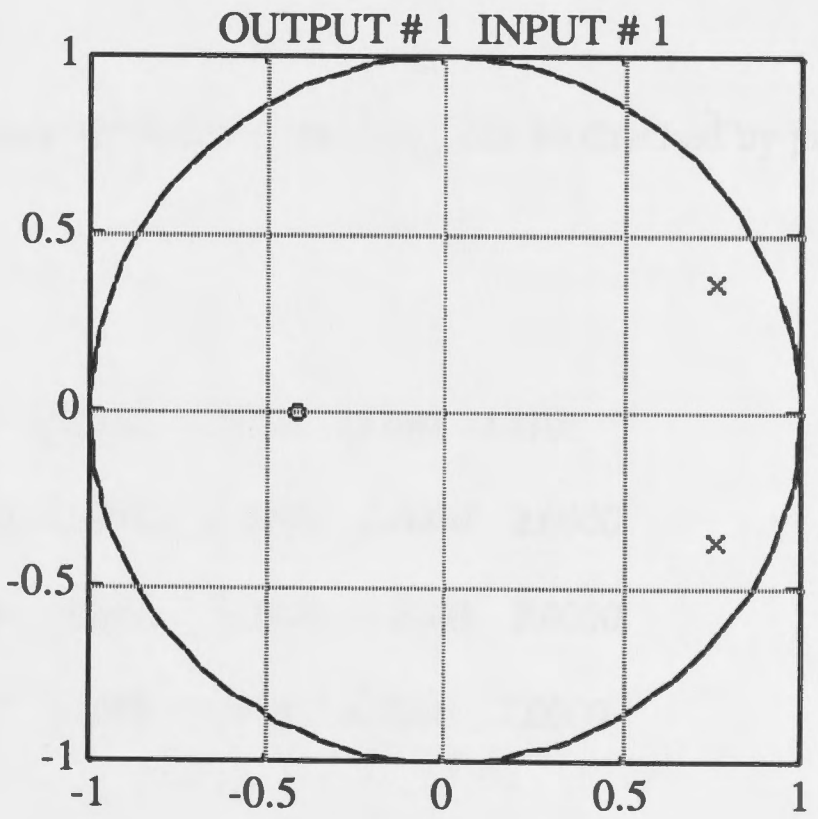


Figure I.13: Zero-pole plot for the ARMAX model

I.3 Model structure determination

This section is based on the same "hairdryer" data in section I.1. The data has been split into two sets, one for the estimation procedure i.e. 1-500 and is denoted by *ze* and the other set for validation purposes, i.e. 501-1 000 and is denoted by *zr*. The data has been detrended and a model of the following kind will be used:

$$y(t) + a1y(t - 1) + \ldots + anay(t - na) = b1u(t - nk) + \ldots bnbu(t - nb - nk + 1) \quad \text{(I.6)}$$

The first step is to determine the time delay (*nk*) by trying out every possible delay between 1 and 10, and then selecting a second order model (*na* = *nb* = 2). The loss function for the different models are computed using the reference data set (*zv*):

```
V = arxstruc (ze, zr, struc(2,2, 1:10));
```

By using the *selstruc* function, the delay that gives the best fit for the reference data set is selected:

```
[nn, Vm] = selstruc(V,0);
```

The chosen structure is given as follows:

```
nn =

     2     2     3
```

The dependence of the fit on the delay can be checked by just typing *Vm*:

```
Vm =

Columns 1 through 7

 -I.7348  -5.9203  -6.4845  -I.8401  -I.5975  -I.5160  -I.4101

   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000

   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000

   1.0000   2.0000   I.0000   I.0000   5.0000   6.0000   7.0000
```

Columns 8 through 10

-I.3974 -I.4326 -I.4426

2.0000 2.0000 2.0000

2.0000 2.0000 2.0000

8.0000 9.0000 10.0000

The logarithms of a quadratic loss function are given as the first row, while the indexes *na, nb* and *nk* are given as a column below the corresponding loss function. The choice of three delays is thus rather clear because it has the lowest loss function.

The next step is to test the orders. The fit for all 25 combinations of up to 5 *a* parameters and up to 5 *b* parameters, all with three delays are checked:

V = arxstruc(ze, zr, struc(1:5, 1:5, 3));

The best fit for the reference data set is obtained for

nn = selstruc(V, 0)

nn =

4 4 3

It is always advisable to check how much the fit has improved with the higher order models. Figure I.14 shows a plot with the fit as a function of the number of parameters used. Note that several different model structures use the same number of parameters. The best structure obtained has a total of 8 parameters, i.e. *nn = [4 4 3]* whilst the loss function for a model with 4 parameters (*nn = [2 2 3]*) is rather comparable.

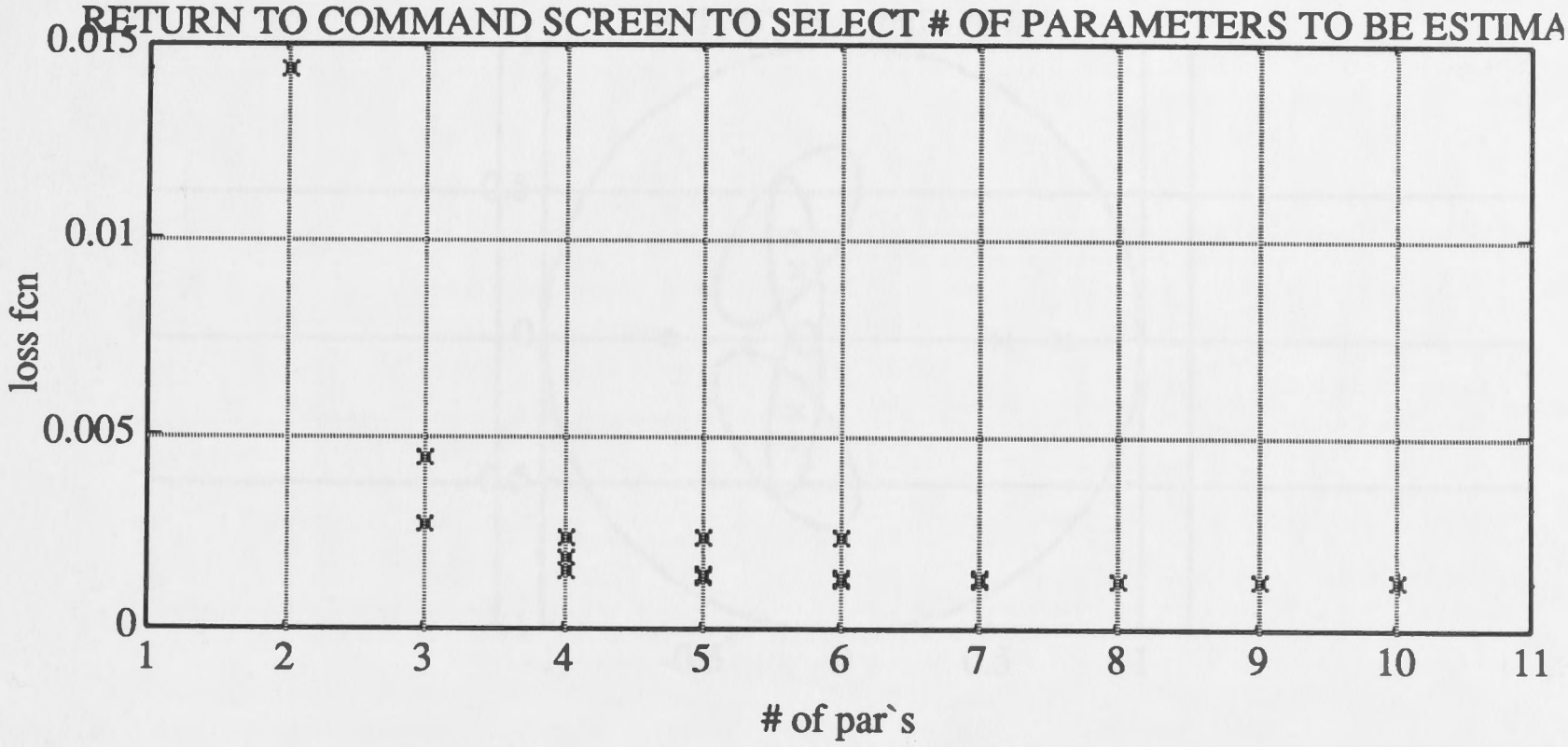


Figure I.14: The values of the loss functions in V against the total number of parameters in the model (MatLab output).

Pole-zero cancellation is useful in determining whether a model is overparameterised or not. The pole-zero configuration for the fourth order model with confidence regions corresponding to three standard deviations is computed as follows (figure I.15):

zpplot(th2zp(th4), 3),

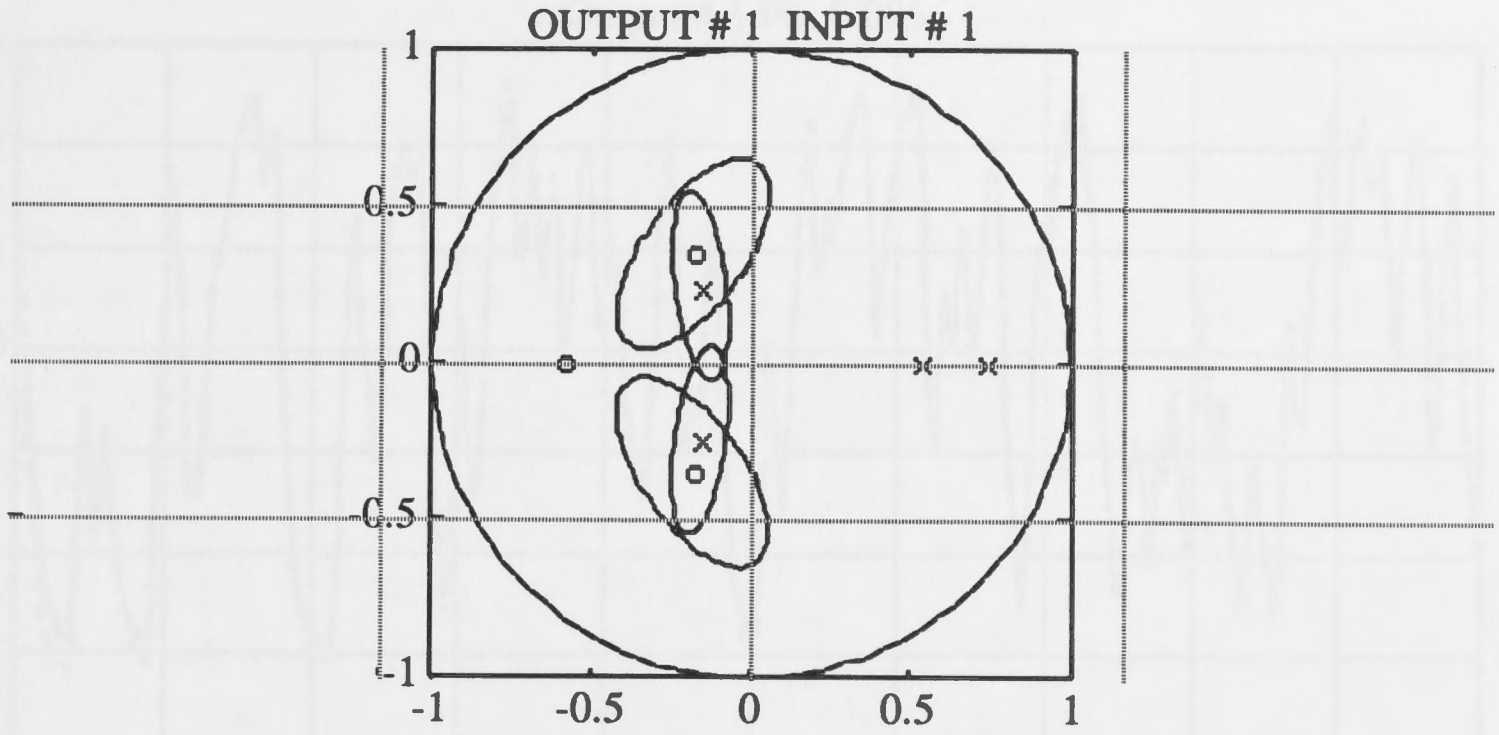


Figure I.15: Zero-pole plot for the fourth order model

It is clear that the two complex conjugated pole-zero pairs cancel; hence a second order model is adequate. A near pole-zero cancellation in the dynamics of a model is an indication that the model order may be too high (Ljung, 1991). To judge if a near cancellation is a real cancellation, the uncertainties in the pole and zero locations are taken into consideration. The function *th2zp* computes confidence regions for the poles and zeros that are graphed, where the integer entry indicates the number of standard deviations for the confidence interval (Ljung, 1991). If the confidence regions overlap, lower model orders should be tried.

Therefore the second order model can be computed:

th2 = arx(ze, [2 2 3]);

The model can be validated for its capability of reproducing the reference data set by comparing the simulated output (from the model) against the observed output (figure I.16).

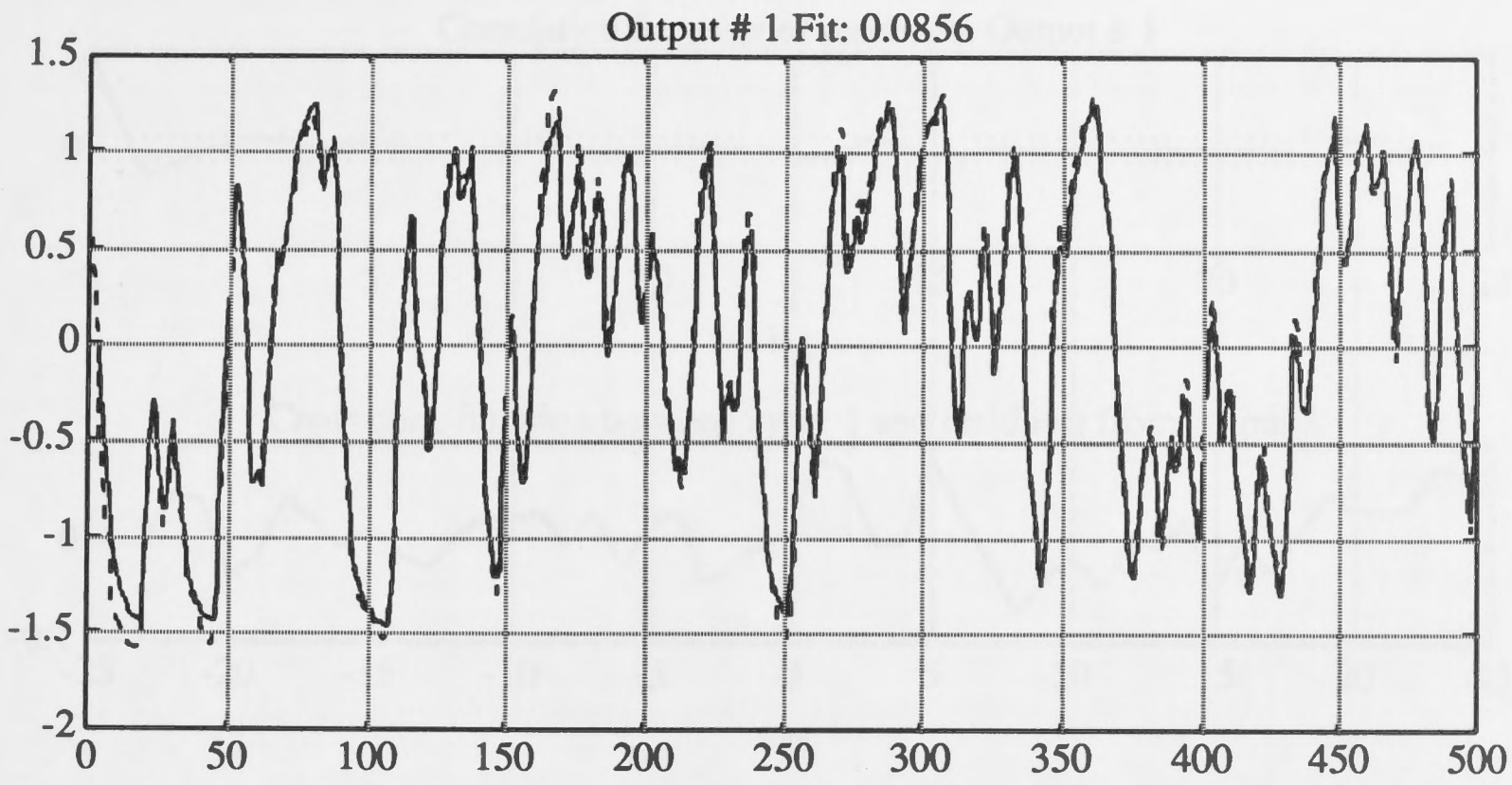


Figure I.16: Comparison of the ARX model with observed output data

The comparison is quite good. The residuals are also checked (figures I.17 and I.18). It is clear that the residuals are quite small compared to the signal level of the output, that they are reasonably (although not perfectly) well uncorrelated with the input and between themselves. Thus model *th2* is satisfactory.

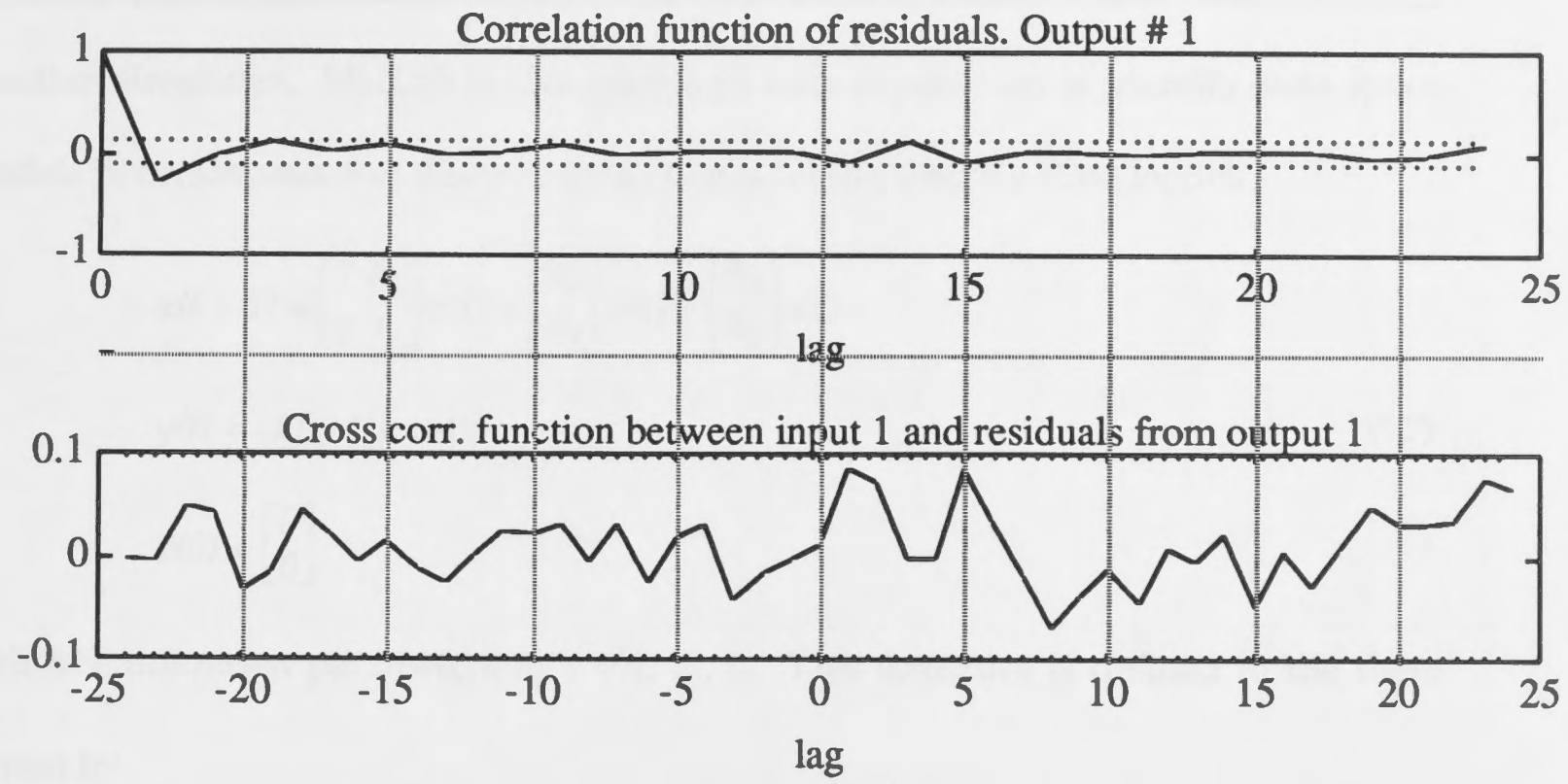


Figure I.17: Autocorrelation function of *e* and cross correlation between *e* and *u* from model obtained by a second order ARX model (MatLab output).

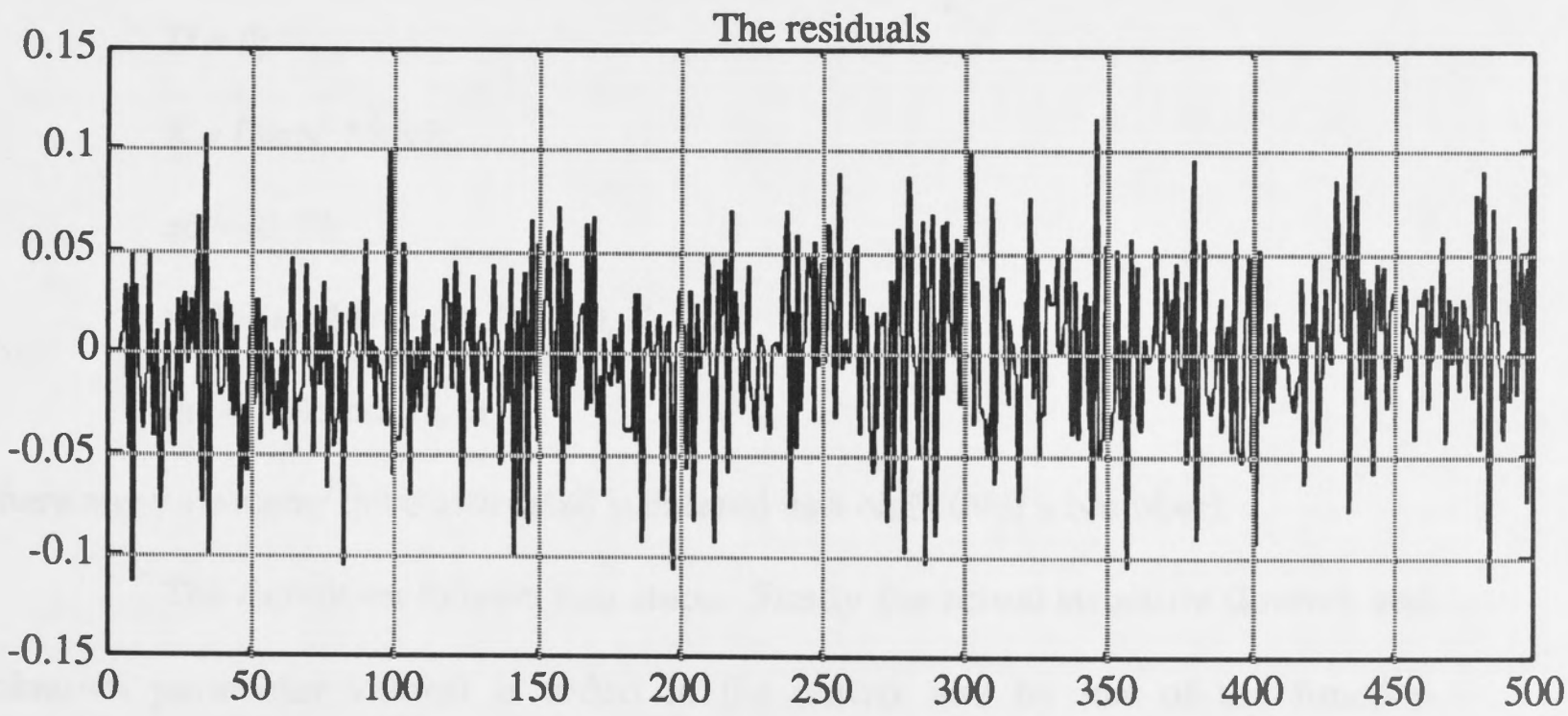


Figure I.18: Plot of the residuals of the second order ARX model

I.4 State space models

State space models have one great advantage in that insight into the physical mechanisms of a system can more easily be incorporated than in the transfer

function structures. MatLab is also equipped with capabilities to identify state space models in continuous and discrete-time. Consider the discrete-time model:

$$x(t+1) = \begin{bmatrix} 1 & \theta_1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} \theta_2 \\ \theta_3 \end{bmatrix} u(t) + \begin{bmatrix} \theta_4 \\ \theta_5 \end{bmatrix} e(t)$$

$$y(t) = [1\ 0]x(t) + e(t) \tag{I.7}$$

$$x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

with five unknown parameters θ_i, $i = 1, \ldots, 5$. This structure is defined in the theta format by

```
A = [1 NaN; 0 1];

B = [NaN; NaN];

C = [1 0];

D = 0;

K = [NaN; NaN];

x0 = [0; 0];

ms1 = modstruc (A, B, C, D, K, x0);

th1 = ms2th (ms1, 'd')
```

where any parameter to be estimated is entered as a NaN (Not a Number).

The definition follows two steps. Firstly the actual structure (known and unknown parameter values) is coded in the matrix *ms1* by use of the function *modstruc*. This function applies to both the continuous and discrete-time cases where the appropriate case is determined or defined as an argument in the function *ms2th*. The theta format matrix is created by *ms2th*. Discrete-time is denoted by the argument '*d*'. Other optional arguments allow the definition of guessed values of the unknown parameters, as well as specification of the covariance matrix of *e(t)* and of the sampling interval *T* (Ljung, 1991).

From here the model's quality can be validated, poles and zeros, and their uncertainties computed.

Endnote IA: Pole-Zero Plots

The z-transform reduces to the Fourier transform when the magnitude of the transformed variable z is unity (i.e. for $z = e^{j\omega}$) that has been simply equated to ω. Thus the z-transform reduces to the Fourier transform on the contour in the complex z-plane, corresponding to a circle with a radius of unity (Oppenheim and Willsky, 1983):

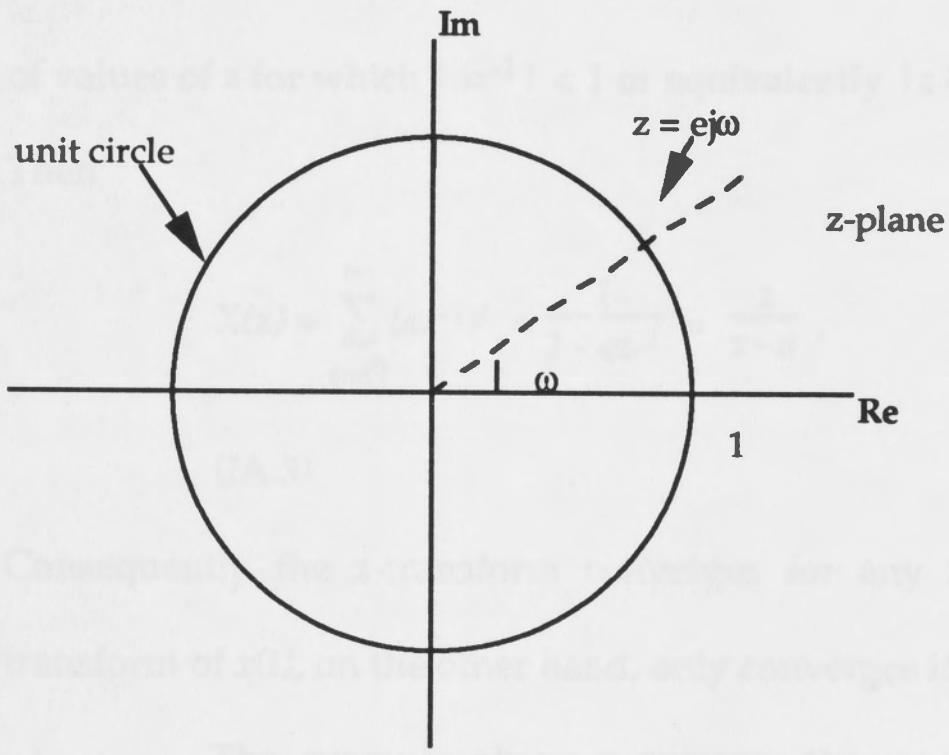


Figure IA.1 : Complex z-plane. The z-transform reduces to the Fourier transform for values of z on the unit circle.

With this change in notation the signal $X(z)$ becomes

$$X(z) \mid_{z = e^{j\omega}} = F\{x[t]\} = X(e^{j\omega}) \qquad \text{(IA.1)}$$

This circle in the z-plane is referred to as the *unit circle*. In general there is associated with the z-transform of a sequence, a range of values of z for which the signal $x(z)$ converges. This range of values is referred to as the *region of convergence* (ROC). If the ROC includes the unit circle, then the Fourier transform also converges (Oppenheim and Willsky, 1983).

Consider the signal $x(t) = a^t u(t)$. Then from the z-transform of a sequence $x(t)$ which is defined as $x(z) \equiv \sum_{t=0}^{\infty} x(t)z^{-t}$, where z is a complex variable,

$$x(z) = \sum_{t=-\infty}^{+\infty} a^t u(t) z^{-t} = \sum_{t=0}^{\infty} (az^{-1})^t \qquad \text{(IA.2)}$$

For convergence of $X(z)$, it is a requirement that $\sum_{t=0}^{\infty} |az^{-1}|^t < \infty$. Thus ROC is the range of values of z for which $|az^{-1}| < 1$ or equivalently $|z| > |a|$.

Then

$$X(z) = \sum_{t=0}^{\infty} (az^{-1})^t = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}, \qquad |z| > |a|$$

(IA.3)

Consequently the z-transform converges for any finite value of a. The Fourier transform of $x(t)$, on the other hand, only converges if $|a| < 1$.

The example above represents the z-transform as a rational function. Consequently it can be characterised by its zeros (the roots of the numerator polynomial) and its poles (roots of the denominator polynomial). There is one zero in the example above at $z = 0$ and one pole at $z = a$.

APPENDIX II

Optimal Control

Optimisation of discrete-time systems is an activity which frequently takes place as one of the central steps in a design process, when solving certain multistage problems. In general, the purpose of this activity is to find a combination of parameter values which will 'best' solve the problem under given (fixed) conditions, i.e. under existing technological and/or ecological restrictions, maximum satisfaction with a limited budget (cost) is sought after. The search for the 'best' parameter values then takes place as an optimisation of the discrete-time model representing the multistage system. Control theory and solution techniques for optimisation of discrete-time systems are the subject of this appendix.

II.1 Historical Account

Techniques available for dealing with multistage optimisation are as follows:

(a) heuristic methods/simple direct methods of calculation;

(b) classical differential calculus;

(c) Lagrangian multiplier method;

(d) calculus of variations;

(e) experimental search method;

(f) dynamic programming (DP); and

(g) Pontryagin's maximum principle (PMP or simply MP) method.

The last two methods are most specialised and yet generalised to cope with the problem of multistage optimisation.

The DP method was founded and developed mainly by Bellman (1957, 1961, 1962). The method was based on the *Principle of Optimality* which was formulated by Bellman (1957) as follows:

> 'An optimal set of decisions has the property that whatever the first decision is, the remaining decisions must be optimal with respect to the outcome which results from the first decision.'

Although the DP method was also extended to continuous-time systems, it found its main application in optimisation of discrete-time systems or multistage optimisation problems. The method is very powerful in treating these problems and its application is mainly limited by the extensive need for computer memory which can happen in some cases; the so called 'curse of dimensionality'.

The maximum principle (MP) was first hypothesised by Pontryagin (1956) and then developed by him and his associates (Boltiyanskii, 1956; Boltiyanskii, et al., 1957, 1958a, 1958b; Pontryagin, 1957, 1959, 1962). The main idea of MP was to construct a special function that depended on the controls and states and was called the Hamiltonian. The Hamiltonian was then optimised by an optimal decision at each stage, given the optimal values of the states. In those days, the calculations were confined to continuous-time systems.

The first attempt to extend MP to the optimisation of discrete-time systems was made by Rozonoer (1959). He concluded that MP was not generally valid for discrete-time systems and applied always to linear discrete-time systems with a linear criterion. Throughout the sixties various authors (Chang, 1960, 1961; Katz, 1962; Fang and Wang, 1964 and so on) did further research to prove the applicability of MP to other special types of discrete-time systems and a major breakthrough only came in 1964 with Halkin and Propoi's work. They showed that MP was valid when the set consisting of all possible values of states and criteria was

convex in each stage. Holtzman (1966) substantially confined this assumption to directional convexity, i.e. convexity with regard to only one direction, which is the direction of increasing the criterion value.

The DP and MP have their weak and strong points. The DP approach is of extremely wide applicability, since it calls for minimum requirements to the functions of the problem: they need neither be differential nor continuous. In this sense, the DP method is very powerful. The price paid for this advantage, is that a lot of combinations of stages and criterion values are to be calculated and stated. Thus even fairly small problems require a great memory capacity and can prove to be prohibitive for a solution (Nahorski, Ravn and Vidal, 1983). MP does not suffer from this weakness. On the contrary, it breaks the problem to a sequence of smaller problems. The drawback is that the functions of the problem must be well-behaved in terms of continuity, differentiability and in many practical cases, convexity or linearity.

Note that in the case of the MP method, the fulfilment of the requirements of directional convexity is not necessary to guarantee a solution and hence the two solution methods have something in common when considering continuous-time problems (Nahorski et al., 1983). In continuous-time the connection between the two methods is seen, for example, in the Hamilton-Jacobi equation that involves functions found both in DP and MP methods.

On the other hand, a clear relationship between MP and nonlinear programming exists. The connecting element is the shadow price in nonlinear programming and the costate vector in the MP algorithm. In numerical methods based on the penalty method, the slope of the penalty function at the optimum point assumes the same value as the costate vector and shadow price. The reason for using

MP instead of nonlinear programming is again the nature of the MP method to reduce the number of variables at each stage of the calculation.

II.2 Control concepts for discrete-time systems

Concepts of *stability, controllability, reachability* and *observability* are useful in understanding the control of discrete-time systems.

II.2.1 Stability

Stability defines the expected behaviour of a system when subjected to perturbations. It occupies a key position in control theory for the reason that the upper limit of the performance of a feedback control system is often set by stability considerations. Thus the designer of a feedback control system will usually push the performance to its limits until stability considerations prevent further improvement. A stable system will respond in some 'reasonable' manner to an applied input. For an unstable system there is little apparent relation between the system input and the system output.

Though determining the stability of a physical system under all conditions is considerably difficult, the case is not so with LTI systems. An LTI system is *bounded-input-bounded-output* (BIBO) stable if a bounded input gives a bounded output for every initial value (Astrom and Wittenmark, 1984). Asymptotic stability is defined in subsection 2.1.1 and for an LTI system, is achieved if and only if all eigenvalues of the system matrix A (equation (3.16a) are strictly inside the unit circle. There are other methods of verifying asymptotic stability that will not be referred to in this appendix (Leigh, 1992).

To establish whether a system is stable or not, it is not required to know the solution, but only to know whether after perturbation the solution decays or

grows. Note that, for a linear system, the responses to initial perturbations of different magnitudes are identical except for a scaling factor.

II.2.2 Controllability and Reachability

Given a finite-dimensional linear system, (II.1a) and (II.1b), there exists a transformation that isolates the controllable and uncontrollable parts (Goodwin and Sin, 1984).

$$x(t+1) = Ax(t) + Bu(t) \tag{II.1a}$$

$$y(t) = Cx(t) \tag{II.1b}$$

The system (II.1a) and (II.1b) is controllable if it is possible to find a control sequence such that the origin can be reached from any initial state in finite time (Astrom and Wittenmark, 1984). A concept related to controllability is reachability. The system (II.1a) and (II.1b) is reachable if it is possible to find a control sequence such that an arbitrary state can be reached from an initial state in finite time.

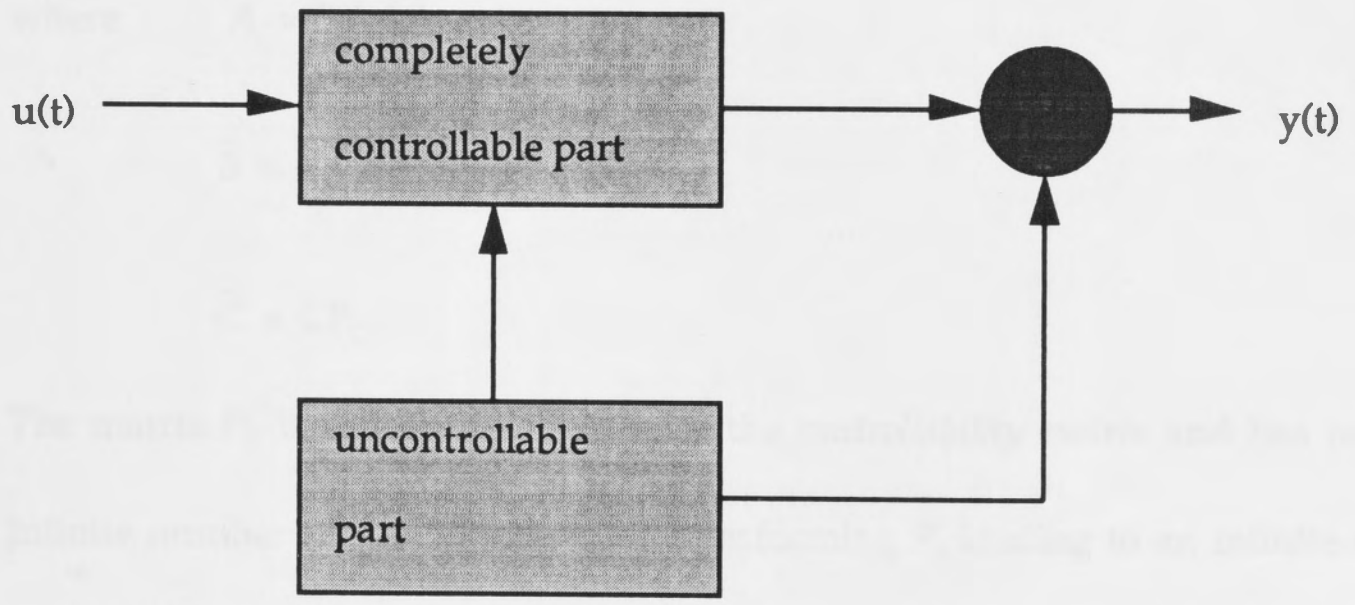


Figure II.2.1 : Decomposition into controllable and uncontrollable parts (Goodwill and Sin, 1984).

As shown in fig. II.2.1 the input signal influences only the completely controllable part of the system. A rational transfer function identifies the controllable part of the

system and its description, using the state space form, would guarantee an observable, controllable and reachable representation. Assuming complete controllability of (II.1) where the initial state, $x(0)$ is given and n is the order of the system, an equivalent representation to (II.1) can be obtained by simply choosing a new basis for the state space.

For example using a transformation such that

$$\bar{x}(t) = P_c^{-1} x(t), \qquad \text{where } P_c \text{ is non-singular} \tag{II.2}$$

and $$P_c^{-1} = [B\ AB\ \ldots\ A^{n-1}B]^{-1} \tag{II.3}$$

giving

$$\bar{x}(t+1) = \bar{A}\bar{x}(t) + \bar{B}\,u(t); \quad \bar{x}(0) = \bar{x}_0 \tag{II.4a}$$

$$y(t) = \bar{C}\bar{x}(t) \tag{II.4b}$$

where $$\bar{A} = P_c^{-1} A P_c$$

$$\bar{B} = P_c^{-1} B$$

$$\bar{C} = C P_c$$

The matrix P_c is usually referred to as the *controllability matrix* and has rank n. An infinite number of choices exists for transforming P, leading to an infinite number of equivalent completely controllable state-space models. Special forms of the state space model can be achieved by forming P in particular ways, for example, by using any n linearly independent columns chosen from the controllability matrix (Goodwin and Sin, 1984). Kailath (1980) expounded on the controllable forms and controller forms that can be created.

If the inverse of P does not exist, the system is said to be uncontrollable and all the poles of the closed-loop system cannot be placed. The following paragraph shows the derivation of P_C:

Suppose that a control sequence to drive the state x in (II.1) from a given state $x(0)$ to a given desired state $x(d)$ for some particular value of t has to be generated.

$$x(1) = Ax(0) + Bu(0)$$

$$x(2) = Ax(1) + Bu(1)$$

$$x(3) = Ax(2) + Bu(2)$$

$$= A(Ax(1) + Bu(1)) + Bu(2)$$

$$= A(A(Ax(0) + Bu(0)) + Bu(1)) + Bu(2)$$

$$= A(A^2x(0) + ABu(0) + Bu(1)) + Bu(2)$$

$$= A^3x(0) + A^2Bu(0) + ABu(1) + Bu(2)$$

$$= A^3x(0) + [B\ AB\ \dots\ A^2B]\begin{bmatrix} u(2) \\ u(1) \\ u(0) \end{bmatrix}$$

and in general

$$x(t) = A^tx(0) + [B\ AB\ \dots\ A^{t-1}B]\begin{bmatrix} u(t-1) \\ \cdot \\ \cdot \\ \cdot \\ u(0) \end{bmatrix}$$

Assuming invertibility

$$\begin{bmatrix} u(t-1) \\ \cdot \\ \cdot \\ \cdot \\ u(0) \end{bmatrix} = [B\ AB\ \dots\ A^{t-1}B]^{-1}(x(t) - A^tx(0))$$

If $x(t)$ is replaced by the desired state, $x(d)$, then an algorithm for generating a control sequence results:

$$\begin{pmatrix} u(t-1) \\ . \\ . \\ . \\ u(0) \end{pmatrix} = [B\ AB\ \ldots\ A^{t-1}B]^{-1}(x(d) - A^t x(0)) \qquad \text{(II.5)}$$

where

$$[B\ AB\ \ldots\ A^{t-1}B]^{-1} = P_c^{-1}$$

If P_C has rank n, then it is possible to find n equations from which the control signals can be found such that the initial state is transferred to the final state $x(n)$.

Note that controllability does not imply reachability. If $Ax(0) = 0$, the origin will be reached with the zero input but the system is not necessarily reachable. The two concepts are, however, equivalent if A is invertible. By the Cayley-Hamilton theorem[1] it is found from (II.1) that all states that can be reached from the origin are spanned by the columns of the controllability matrix P. This implies that the reachable states belong to the linear subspace spanned by the columns of P.

II.2.3 Observability

The system (II.1) is observable if there is a finite t such that knowledge of the inputs $u(0), \ldots, u(t-1)$ and the outputs $y(0), \ldots, y(t-1)$ is sufficient to determine the initial state of the system (Astrom and Wittenmark, 1984). Assuming $y(0), y(1), \ldots, y(n-1)$ are given, the following set of equations are obtained:

---

[1] Cayley-Hamilton theorem: Every matrix is a zero of its characteristic polynomial (Lipschutz, 1987)

$$y(0) = Cx(0)$$

$$y(1) = Cx(1) = CAx(0)$$

$$\vdots$$

$$y(n-1) = CA^{n-1}x(0)$$

Using vector notation,

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} x(0) = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(n-1) \end{bmatrix} \qquad \text{(II.6)}$$

The state $x(0)$ can be obtained from (II.6) if and only if the *observability matrix*,

$$P_o = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(n-1) \end{bmatrix} \text{ has rank } n.$$

The state $x(0)$ is unobservable if $x(0)$ is in the null space of P_o. If two states are observable, then any linear combination is also unobservable, i.e. the unobservable states form a linear subspace.

It can be shown that the controllability matrix P_c is the transpose of the observability matrix, i.e. $P_o \triangleq P_c^{-1}$ (Goodwin and Sin, 1984) . Thus P_o can be directly determined from the original observability matrix by searching it by rows, constructing a matrix Q from the rows that are selected and finally putting $P_o = Q^{-1}$.

Note that $Q = P_c^{-1}$ since Q is constructed by rows, whereas P_c is constructed by columns.

II.3 Multistage decision processes

A multistage decision maybe considered as an abstract notion by which a large number of activities can be represented. A process is either deterministic or stochastic. Since a multistage decision process is an entity consisting of a finite number of stages, its nature is completely determined by the types of stages from which it is formed and by the ways the stages are interconnected.

A stage can represent any real or abstract entity, e.g. a space unit, a time period, or an economic activity in which a certain transformation takes place. Those variables that are transformed in each stage are called state variables. The desired transformation for the state variables is achieved through manipulation of decision variables that remain or may be considered to remain constant within each stage of the process. The transformation at each stage is completely described by a set of performance equations. A stage may have any number of input and output streams by which the state variables are transferred into and out of the stage.

Continuous *control processes*[2] can be discretised. When a continuous control process is viewed in this way, it takes on the characteristics of a multistage decision process (Elgerd, 1967).

---

2 The pair of vector functions ($u(t)$, $x(t)$), that is, the control $u(t)$ and the corresponding phase trajectory $x(t)$, will be called, in this dissertation, the *control process* or simply *process* (Boltyanskii, 1971). A process can be categorised either as a homogeneous or a heterogeneous process depending on the form of the performance equations. For a homogeneous process the state variables and the decision variables are interrelated by the same set of performance equations throughout the process.

The problem of optimising a multistage decision process is called a multistage optimisation problem. The objective function of the process, which is to be maximised or minimised, can be expressed as a function of the state variables leaving the last stage of the process. Thus a general multistage optimisation problem can be stated as follows:

> For a process with all the performance equations and the initial and/or final values of some of the state variables given, find the values of the decision variables at each stage, subject to certain constraints in such a way that the objective function is maximised or minimised.

In transferring the state of a system from x_0 to x_f in t_f time units, an infinite number of trajectories exist (figure II.3.1(a)), and only one of these corresponds to a minimum value. In *static optimisation* the *Euler-Lagrange* equation presents the right one. Figure II.3.1(b) shows the descretised version of the same process. The control period is divided into N equal stages and the process unfolds in the following manner:

From stage *0* to stage *1* to stage *2*, and so on, until the final stage or destination. At each of the stages *0, 1, ..., N-2*, a decision must be made about which of the possible routes should be chosen. Suppose these decisions are given by q_0, q_1, ..., q_{N-2} and are defined by a decision vector **q**.

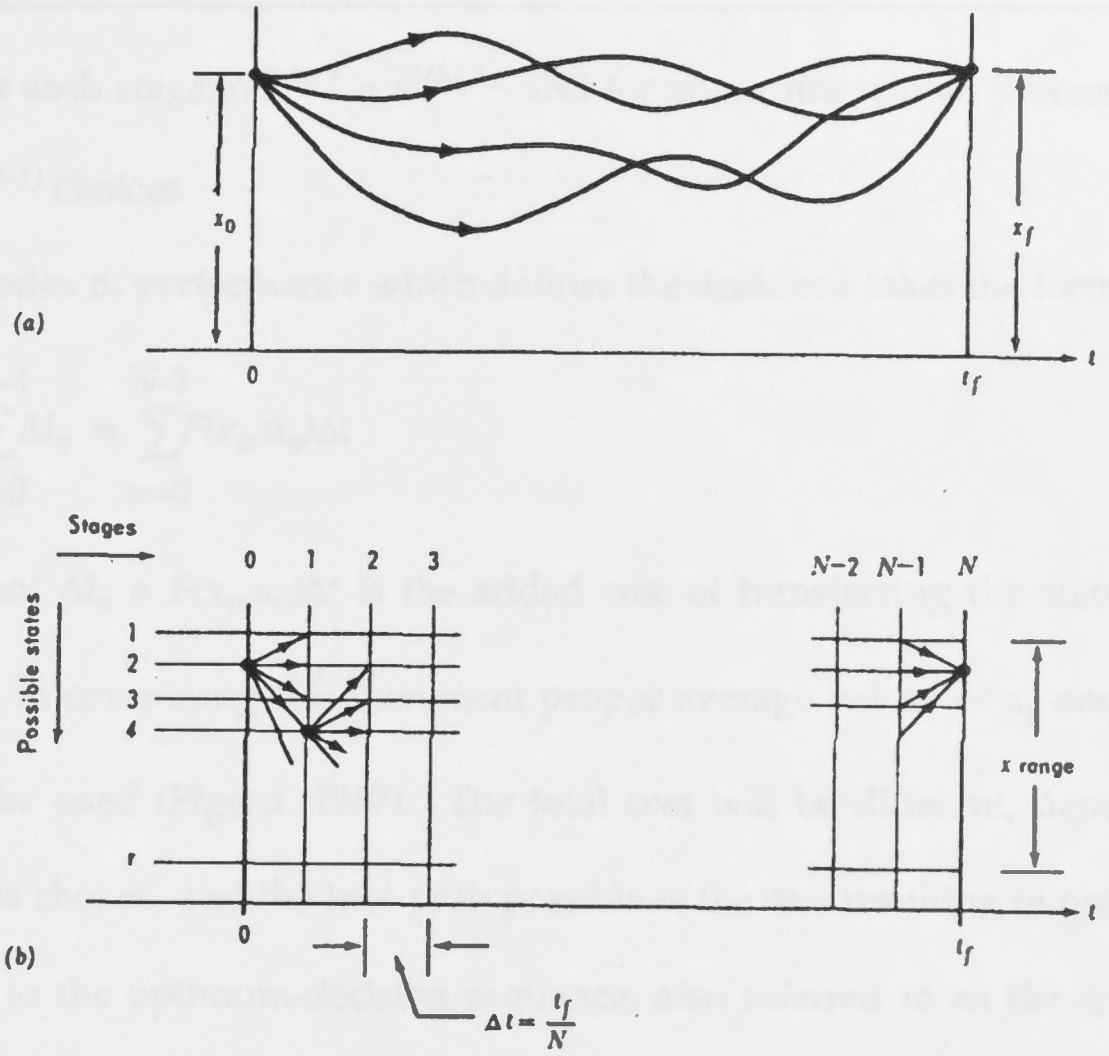


Figure II.3.1: Multi-stage Process (Elgerd, 1967)

Note that when, after *N-1* decisions, stage *N-1* is reached, no further choice is made as a consequence of the specification of the end state. The range of possible states (marked 'x range' in figure II.3.1) must be determined from the physics of the problem. The range may be narrowed from the outset, which would be for the better.

The **q** vector uniquely determines the overall path taken. The choice of the coarseness of the grid, i.e. magnitudes of the numbers N and r, will determine the eventual accuracy of the analysis (Elgerd, 1967) and clearly N and r should be as large as possible. In going from $0 \rightarrow 1$, there are r choices or possibilities and the two-stage process of going from $0 \rightarrow 1 \rightarrow 2$ involves r^2 possibilities. Therefore the total N-stage process involves r^{N-1} possibilities, but in only one dimension. For a two dimensional process involving 2 state components x_1 and x_2, assuming that each has r different

discrete values at each stage, $(r^2)^{N-1} = r^{2(N-1)}$ and for an n-dimensional process there are therefore $r^{n(N-1)}$ choices.

The index of performance which defines the decisions takes the form

$$I = \sum_{v=0}^{N-1} \Delta I_v = \sum_{v=0}^{N-1} F(x_v, u_v)\Delta t \qquad \text{(II.7)}$$

The cost increment $\Delta I_v = F(x_v, u_v)\Delta t$ is the added cost of transferring the state from stage v to $v + 1$. In computing this increment proper average values of x_v and u_v in the interval can be used (Elgerd, 1967). The total cost will be different, depending upon what path is chosen, and the best path possible is the one resulting in optimum I, corresponding to the optimum-decision sequence, also referred to as the *optimum policy*.

In selecting the optimal policy, the obvious approach would be to compute the total cost index I along all possible path combinations and then choose the best one. Sometimes this can be time consuming depending on the scale of the process even if the computation is being done on a fast computer.

II.4 Dynamic programming

Consider a system governed by a recursion equation of the form:

$$x(t + 1) = f(x(t), u(t), t) \qquad \text{(II.8)}$$

where

$x(t)$ = state of the system at instant t

$u(t)$ = control at instant t.

The sequence $x(0), x(1), x(2), \ldots, x(t)$ is called the trajectory of the system associated with the control sequence $u(0), u(1), u(2), \ldots, u(t)$. Consequently every state sequence is called a permitted trajectory if it can be realised by one or more controls. The

number of steps involved is called the horizon. This number is assumed to be finite. The problem is to find the control sequence that maximises a criterion of the form

$$v_0(x(0), u(0)) + v_1(x(1), u(1)) + \ldots + v_{N-1}(x(N-1), u(N-1)) + V_N(x(N)) \qquad \text{(II.9)}$$

for a given initial state.

Assuming the principle of optimality, if a control sequence $\tilde{u}(0)$, $\tilde{u}(1)$, ..., $\tilde{u}(N-1)$ is optimal, so is the control $\tilde{u}(t)$, ..., $\tilde{u}(N-1)$ for the initial state $\tilde{x}(t)$ and the criterion,

$$v_t(\tilde{x}(t) , u(t)) + v_{t+1}(x(t+1), u(t+1)) + \ldots + v_{N-1}(x(N-1), u(N-1)) + V_N(x(N)). \qquad \text{(II.10)}$$

Setting

$$V_{N-1}(x(N-1)) = \max_{u(N-1)} v_{N-1}(x(N-1), u(N-1) + V_N(x(N)), \qquad \text{(II.11)}$$

in general,

$$V_t(x(t)) = \max_{u(t)} v_t(x(t), u(t) + V_{t+1}(x(t+1)), \qquad \text{for } t = 0, \ldots, N-1. \qquad \text{(II.12)}$$

The above equations are called the *realisation equations*.

A necessary and sufficient condition for a control $\tilde{u}(0)$, $\tilde{u}(1)$, ..., $\tilde{u}(N-1)$ to be maximal is that

$$V_t(\tilde{x}(t)) = v_t(\tilde{x}(t) , \tilde{u}(t) + V_{t+1}(\tilde{x}(t+1)), \qquad (\forall t = 0, \ldots, N-1). \qquad \text{(II.13)}$$

If a control $\tilde{u}(0)$, $\tilde{u}(1)$, ..., $\tilde{u}(N-1)$ is maximal, then the control $\tilde{u}(0)$, $\tilde{u}(1)$, ..., $\tilde{u}(t-1)$ is maximal (the initial state remaining unchanged) for the criterion

$$v_0(x(0), u(0) + \ldots + V_{t-1}(x(t-1)) + V_t(x(t)) \qquad \text{(II.14)}$$

Conversely, if $\tilde{u}(0)$, $\tilde{u}(1)$, ..., $\tilde{u}(t-1)$ is maximal for this criterion, it can be extended as an optimal control for the initial criterion.

In the case in which the criterion needs to be minimised, the 'max' is replaced by 'min'. A case dealing with an infinite horizon is dealt with quite extensively by de la Barriè re (1967; p 333-337) and is very important in determining the optimal stand density where it is critical to know the length of the plantation

rotation other than just the number of thinnings, intensity of thinnings and timing of thinning.

II.5 Discrete maximum principle

The process described in this section is a generalised algorithm capable of obtaining directly the optimal policy for the entire process without decomposing the process into sub processes which are optimised independently.

Given all the performance equations for a process, it is desired to find the value of a decision vector at each stage, subject to a certain function of the state vector leaving the last stages. All the values of the state vector entering the initial stages and leaving the last stages are unspecified. The desired transformation for the state variables is achieved through manipulation of decision variables which remain or may be considered to remain constant within each stage of the process.

A stage may have any number of entering and leaving streams by which the state variables are transferred into and out of the stage. It is convenient to classify the stages by the number of streams with which they are associated. There are four basic types which are:

(a) linking stage - one entering stream and one leaving stream;

(b) separating stage - one entering stream and several leaving streams;

(c) combining stage - several entering streams and one leaving stream; and

(d) complex stage - several entering streams and several leaving streams.

Only type (a) will be considered in this section because of the nature of the process models dealt with in this dissertation, i.e. SISO (single-input/single-output) models.

Let an s-dimensional vector x represent the state vector, and a n-dimensional vector θ , the decision vector. The set of performance equations for a linking stage is of the form:

$$x(t) = T(t)(x(t\text{-}1), \theta(t)) \tag{II.15}$$

where

$T(t)$ = transformation operator.

To find the optimal sequence of the decision vector, a covariant vector z and a Hamiltonian function H are introduced for a linkage stage,

$$z(t\text{-}1) = \frac{\partial H(t)}{\partial x(t-1)} \tag{II.16}$$

$$H(t) = \sum_{i=1}^{s} z(t)_i T(t)_i (x(t\text{-}1), \theta(t)) \tag{II.17}$$

The optimal decisions at these stages are then determined by the following conditions:

$$\frac{\partial H(t)}{\partial \theta(t)} = 0 \text{ or } H(t) = \text{maximum} \tag{II.18}$$

In the maximisation of the Hamiltonian function, both x and z are considered as fixed at each stage. The value of $z(0)_i$ and $z(N)_i$ are determined by

$$z(0)_i = 0 \text{ or } x(0)_i z(0)_i = \text{maximum} \tag{II.19}$$

$$z(N)_i = \frac{\partial \phi(x(N))}{\partial x(N)_i} \tag{II.20}$$

where $\phi(x(N))$ is the objective function.

From the definition of the covariant vector, it can be seen that there is always one component of z corresponding to one component of x. The specifications

of the values of $z(0)_i$ and $z(N)_i$ are also dependent on those of x. Thus to solve the problem in which the initial and/or final values of some x_i are prescribed, the conditions given in equations (II.19) and (II.20) for the corresponding $z(0)_i$ and $z(N)_i$ should be deleted.

II.6 General remarks on the MP and DP

It is widely recognised that there is no single mathematical optimisation technique superior to all other techniques in handling every type of problem. Every method has its own merits and shortcomings; consequently, it may be suitable in solving some types of problems but become cumbersome in solving others.

Detailed derivation of DP from MP and vice versa, have been presented and conclusions have been made that the two solution techniques are essentially the same (Fan and Wang, 1964). However, it is important to note that the ultimate objective of optimisation is not the formulations of the problems and the methods for solving them, but is the numerical solution of the optimal policy which can be immediately used. Although DP and MP can be derived from each other, which obviously must be true or otherwise one of them would be incorrect, the ways of approach to the problems according to each method are quite different. 'DP will start the investigation by searching the entire grid of the n variables at one stage, store this grid of values, and proceed stage by stage; MP will start the investigation by computing one optimum path along the m-stages and then proceed to improve this optimum path based on the values obtained from the preceding computation.', (Lee, 1963).

As mentioned earlier DP has large memory requirements, a difficulty that is avoided in MP at the cost of introducing covariant variables z. The values $x(N)$ or $z(N)$ must be guessed before starting the computation. In dealing with a complex

process by MP, the number of $x(N)$ or $z(N)$ to be guessed is increased. This corresponds to the increase in dimensionality in DP. However, the increase in the number of guesses will simply increase the computing time linearly. Consequently, as long as the computer memory capacity is limited, the advantage of the MP cannot be overemphasized.

Nevertheless, the definite and significant advantage of DP in the optimisation of processes with constraints on state variables cannot be neglected. *The processes with bounded state variables do not give any trouble to the method of DP since, in this method, the optimal decisions are determined for the whole allowable domain of the state variables.* Therefore, the optimal policy thus obtained automatically satisfies the constraints on the state variables. Because of the elegant simplicity and versatility of the principle of optimality, the method of DP can be used to handle the processes for which the transformation at each stage is difficult to be expressed in finite difference equations and thus is verbally described. The MP, on the other hand, is applicable only to the processes with well-defined performance equations, and the transformation functions must be continuously differentiable with respect to the state variables.

If an optimisation problem has several local optimal policies, there is no assurance that the optimal policy found by MP is the global optimal policy (Fan and Wang, 1964). Since the method of DP employs the so-called *imbedding technique,* which is similar to the exhaustive search, the optimal policy obtained is always the global optimal policy, provided that the interpolation error inherent to the method is negligible. It may be proposed that when an accurate solution is desired or required, the method of DP be employed first to locate approximately the position of global maximum and the MP be applied to pinpoint the maximum point (Fan and Wang, 1964).

II.7 Optimal design method: state space approach

The synthesis problem is formulated to minimise a criterion which is a quadratic function of the states and the control signals. The Linear Quadratic (LQ) controller for the LTI systems leads to a control law of the same structure as the *state feedback controller* (otherwise called *policy* or *decision function*) and can also be interpreted as a pole-placement controller. The controller issues *commands* that are physically connected to a process with the intention to influence the behaviour of the process in a particular way. In feedback control the controller is *error driven*, i.e. the controller receives a continuous measurement of the difference between required behaviour and observed behaviour and its output is some function of this error (Leigh, 1992). Optimal control strategy can be separated into two parts:

(a) state estimator, which gives the best estimates of the states from observed outputs; and

(b) linear feedback law from the estimated states.

Problem Formulation

The design problem is specified by giving the *process*, *criterion*, and *admissible control signals* (Astrom and Wittenmark, 1984).

The process

The sampled model of an LTI system is represented by (3.25) where w and e are discrete-time Gaussian white-noise processes and zero mean value, and

$$Ew(t)\, w^T(t) = R_1 \qquad \text{(see footnote}^3\text{ for } R_1 \text{ expression)} \tag{II.22}$$

$$Ew(t)e^T(t) = R_{12} \tag{II.23}$$

$$Ee(t)e^T(t) = R_2 \tag{II.24}$$

It is also assumed that the initial state $x(0)$ is Gaussian distributed with

$$Ex(0) = m_0 \text{ and } cov(x(0)) = R_0 \tag{II.25}$$

The matrices R_0, R_1 and R_2 are positive semidefinite. It is also assumed that (3.34) is reachable and observable.

Criterion

The purpose of the control is to minimise the loss function when the process is described by (3.16) (Astrom and Wittenmark, 1984):

$$J = E\left\{\int_0^{Nh} [x^T(th)Q_{1c}x(kh) + 2x^T(th)Q_{12c}u(th) + u^T(th)Q_{2c}u(th)] + x^T(Nh)Q_{0c}x(Nh)\right\} \tag{II.26}$$

where the matrices Q_{0c}, Q_{1c} and Q_{2c} are symmetric and positive definite. The matrices in the loss function may depend on time.

Admissible control laws

Periodic sampling is mandatory and the control signal is assumed constant over the sampling periods. If C is the unit matrix and if $v(t) = 0$ in (3.16) then

---

[3] Consider the random variable, $w(t_i) = \int_{t_i}^{t_{i+1}} e^{A(t_{i+1}-s)} dw(s)$. This variable has zero mean and the covariance is given by $Ew(t)\, w^T(t) =$

$$E\int_{t_i}^{t_{i+1}} e^{A(t_{i+1}-s)} dw(s)\, dw^T(t) e^{A(t_{i+1}-t)}$$ (Astrom and Wittenmark, 1984) .

the full state vector is available. The control signal is then allowed to be a function of the state up to and including time *th*. This is called *complete state information*. In most cases state variables are not known exactly. This is called *incomplete state information*.

The problem

The optimal control problem is now defined as finding the admissible control signal (or control sequence) that minimises the loss function of (II.26) when the process is described by (3.16) such that a transfer to a preassigned terminal state *x(f)* (different from *x(0)*) is achieved. More over, it is usually required that the transition process (i.e. the process of transition from the initial state *x(0)* to the preassigned state *x(f)*) be best in a definite sense, e.g. that the transition be accomplished in the shortest time or that minimum energy be expended during the transition process and so on (Boltiyanskii, 1971).

A best transition process is called an *optimal process*. For the shortest transition time, the process is called a *time-optimal process*.

Sampling of the Loss Function

The loss function in (II.26) is expressed in continuous time. The optimal control problem can be transformed into a discrete-time problem of minimising the loss function when the process is described by (3.34) (Astrom and Wittenmark, 1984):

$$J = E\left\{\sum_{t=0}^{N-1}[x^T(th)Q_1x(th) + 2x^T(th)Q_{12}u(th) + u^T(th)Q_cu(th)]\right.$$

$$\left. + x^T(Nh)Q_0x(Nh)\right\} \qquad \text{(II.27)}$$

where

$$Q_0 = Q_{0c} \qquad \text{(II.28)}$$

$$Q_1 = \int_{th}^{th+h} A^T(s, th)\, Q_{1c} A(s, th) ds \tag{II.29}$$

$$Q_{12} = \int_{th}^{th+h} A^T(s, th)[Q_{1c} B(s, th) + Q_{12c}] ds \tag{II.30}$$

$$Q_2 = \int_{th}^{th+h} [B^T(s, th) Q_{1c} B(s, th) + 2B^T(s, th) Q_{12c} + Q_{2c}] ds \tag{II.31}$$

It is further assumed that $Q_{12} = 0$, and can be eliminated. Notice that the sampled loss function (II.27) will have a cross-coupling term, Q_{12} even if $Q_{12c} = 0$.

When the stochastic case is being considered, one additional term depending on the noise is obtained in (II.27). However, this term is independent of the control signal and can thus be disregarded when performing minimisation.

Transformation of the loss function

A transformation is done to simplify the writing by introducing a new control signal

$$\tilde{u} = u + M^T x \tag{II.32}$$

where

$$M = Q_{12} Q_2^{-1} \tag{II.33}$$

The system of (3.34) will be transformed to

$$x(t + 1) = \tilde{A}\, x(t) + B\tilde{u}(t) + w(t) \tag{II.34a}$$

$$y(t) = Cx(t) + e(t) \tag{II.34b}$$

where

$$\tilde{A} = A - BM^T \tag{II.35}$$

Further the loss function of (II.11) is transformed to

$$J = E\left\{\sum_{t=0}^{N-1}[x^T(t)\tilde{Q}_1 x(t) + \tilde{u}^T(t)Q_c\,\tilde{u}(t)] + x^T(N)Q_0 x(N)\right\} \tag{II.36}$$

where

$$\tilde{Q}_1 = Q_1 - Q_{12}Q_2^{-1}\,Q_{12}^T \tag{II.37}$$

II.7.1 Linear Quadratic Control

The LQ problem can now be solved for the case of complete state information and the solution is obtained by using DP.

Deterministic case

Assuming that $w(t) = 0$ and $v(t) = 0$ in (3.34), the system is then described as:

$$x(t+1) = Ax(t) + Bu(t) \tag{II.38a}$$

$$y(t) = Cx(t) \tag{II.38b}$$

where $x(0)$ is given. The problem is to determine the sequence $u(0), u(1), \ldots, u(N-1)$ such that the loss function (II.26) is minimised. The solution is given as follows:

Consider (II.38a and II.38b) and let $u(t)$ be a function of $x(t), x(t-1), \ldots$. Introduce

$$S(t) = A^TS(t+1)A + Q_1 - L^T(t)(Q_2 + B^TS(t+1)B)L(t)$$

$$= [A - BL]^TS(t+1)A + Q_1$$

$$= [A - BL]^TS(t+1)[A - BL] + Q_1 + L^TQ_2L \tag{II.39}$$

where the matrix L is defined by

$$L(t) = (Q_2 + B^TS(t+1)B)^{-1}B^TS(t+1)A \tag{II.40}$$

and the end condition $S(N) = Q_0$.

Now, assuming $S(t)$ has a positive semidefinite solution and that $Q_2 + B^T S(t)B$ is positive definite, then there exists a unique admissible control strategy,

$$u(t) = -L(t)x(t) \tag{II.41}$$

that minimises the loss function (II.26) when $Q_{12} = 0$. The minimum loss is given by

$$\min J = V_0 = x^T(0)S(0)x(0) \tag{II.42}$$

It is important to realise that the mode of operation of the control law (II.41) implies *information gathering* during the control process. The information received by the controller is the value of the current state at each time. Furthermore, this information is utilised directly during the control process since the control at time t depends on the current state $x(t)$ via the function $L(t)$.

Proof:

Introduce

$$V_t = \min_{u(t) \ldots u(N-1)} E\left\{ \sum_{i=t}^{N-1} [x^T(i)Q_1x(i) + u^T(i)Q_2u(i)] + x^T(N)Q_0x(N) \right\}$$

For $t = N$,

$$V_N = x^T(N)S(N)x(N)$$

where

$$S(N) = Q_0$$

For $t = N - 1$,

$$V_{N-1} = \min_{u(N-1)} \{ x^T(N-1)Q_1x(N-1) + u^T(N-1)Q_2u(N-1) + V_N \} \tag{II.43}$$

Using (3.34) gives

$$V_{N-1} = \min_{u(N-1)} \{ x^T(N-1)Q_1x(N-1) + u^T(N-1)Q_2u(N-1)$$

$$+ [Ax(N-1) + Bu(N-1)]^T S(N)[Ax(N-1) + Bu(N-1)] \} \tag{II.44}$$

The control law is given by

$$u(N-1) = -L(N-1)x(N-1) \tag{II.45}$$

with the minimum loss

$$V_{N-1} = x^T(N-1)S(N-1)x(N-1) \tag{II.46}$$

where

$$S(N-1) = A^TS(N)A + Q_1 - L^T(N-1)\,(Q_2 + B^TS(N)B)L(N-1) \tag{II.47}$$

and

$$L(N-1) = (Q_2 + B^TS(N)B)^{-1}B^TS(N)A \tag{II.48}$$

The above arguments can be followed for V_{n-2} with the time arguments in (II.43) shifted one step. The procedure can now be repeated, and V_0, which is the minimum of J, is obtained by iterating backwards in time. The calculations needed to determine the LQ-controller can be made by hand only for very simple examples. In practice it is necessary to have access to interactive programs, which can compute the control law and simulate the systems. Anderson and Moore (1989; p364) outlines the specialisation of MP equations to the linear quadratic problem.

II.8 DP in the determination of Optimal Stand Density for Intensively managed Plantation Forests.

Bellman's DP has gained considerable popularity. As its name implies, it is basically an ingenious method of computer programming used to solve problems that require sequential decision-making. The reference in this section to DP is based on its current use in the Forestry discipline as explained in chapter one.

DP has been extensively applied in areas of inventory and production decisions, allocation and control problems, and in systems design (Bellman 1957; Nemhauser, 1966; Wagner, 1975). In forestry, however, DP has been used sparingly. Arimizu (1958) used it to regulate intermediate cutting with the objective of producing

a maximum harvest volume. Hool (1965), using simply a 'cut' or 'do not cut' strategy, applied a DP model. Later he introduced a Markov chain approach to production control using a DP model (Hool, 1966).

Amidon and Akin (1968) compared traditional marginal analysis with DP for determining optimal growing stock and found the latter to be more flexible and convenient. Other authors have illustrated the feasibility of DP for deriving optimal cutting schedules for timber stands (Risvand 1969; Kilkki and Vaisanen 1969; Schreuder, 1971). Many researchers in forest management studied DP to support the sequential decision making required for decisions about the thinning regime and rotation of even-aged stands (Brodie and Kao, 1979; Chen et al., 1980; Martin and Ek, 1981; Haight, Brodie and Dahms, 1985).

Unfortunately many of the above papers are difficult to follow because explicit derivation of the solution procedures is lacking (Chen et al., 1980). An additional shortcoming of several of the papers is the absence of suitable *forest growth models* - ones directly related to the decision variable. These two factors, plus the unfamiliarity of most readers with the special conditions which must be met for a problem to be solved as a DP problem, account for the limited application of DP in forestry (Chen et al., 1980).

A report on the Methodology to Evaluate Plantation Management Alternatives (Pienaar and Harrison, 1994) for *P. patula* in South Africa, pointed out that the determination of optimal stand density for plantation forestry had not been achieved as yet: 'When management strategies include thinnings the added complexity of the decision problem is such that it defies explicit solution by any available methodology. A few such attempts have been reported in the scientific literature. Naslund (1969) used Pontryagin's (1962) maximum principle and methods of optimal control theory, as did Clark (1976). Kilkki and Vaisanen (1969) and

Schreuder (1968) use a mathematical (dynamic) programming formulation to determine optimum rotation age and thinning strategies. It is our conclusion, based on these examples, that the utility of these optimisation procedures in solving realistic, practical problems of the kind we hope to consider for intensively managed plantations, beyond the use of lumped parameter models, has not been demonstrated.' However, the report did not indicate as to why stand density optimisation has not been a success.

Chen et al., (1980) expounded further on the difficulties foresters have faced in using DP: 'Potential users of DP in forestry have not known how to formulate problems so that DP can be used to solve them. Part of the difficulty may stem from the DP symbolism - the shorthand used in expressing problems in a multistage decision process. Another source of confusion is the variety of decision process frameworks: deterministic, stochastic, finite time, infinite time, discrete state, continuous time, etc.'

Endnote IIA illustrates a typical example of a forestry DP formulation for a thinning problem. It is clear from Endnote IIA that the formulation is rigid in that all the possible states at each stage are specified and that the stages are set at equally spaced intervals. This severely limits the optimisation and also the application of the result. A properly formulated problem with the appropriate models should demonstrate DP's ability to search for a control sequence. At most, the search should be done on a unit time basis, and at each stage perform an exhaustive search for an optimal state variable. There should be an infinite selection of state variables that are limited by the dynamics of the system and in some cases, by user-specified upper and/or lower limits of the control sequence. The control limits need not be constant through the search but can be varied to suit management and ecological constraints. This was demonstrated in chapter five with a real forest thinning problem.

Endnote IIA: A typical example of a Forestry DP formulation for a thinning problem

The control problem dealt with in this book involves the development of a plan to cut an even-aged pine forest stand in such a way as to maximise the total yield or net present value (NPV), from thinning and final harvest. To simplify the explanation of one of the procedures (maximising the total yield) only three thinnings are considered and decisions are to be made as to how much to thin over a time period of 40 years. Three possibilities are considered, i.e. no thinning, a light thinning that removes approximately 12 m^2/ha (basal area), or a heavy thinning that removes approximately 26 m^2/ha.

The time when a decision is made is called a stage. The condition of the stand at a particular stage, just before the decision, is called a state. The consequence of a decision is to move the stand from a particular stage and state to a different state at the subsequent stage. The change is due in part to the thinning and in part to the growth of the trees that remain after thinning. Each decision is associated with an immediate yield (volume - m^3/ha).

A path through the network represents one possible way to manage the stand. Clearly, the best management could be found by following all possible paths through the network and finding the one with the largest total yield. One intuitive way of solving the thinning problem would be to choose at each stage, starting with stage 1, the decision that gives the largest immediate yield. The decision of how much to thin at each stage cannot be made independent of what has been done earlier, or what will be done later. A correct solution must take into account this dependence between decisions at various stages.

Consider the possible decisions at stage *i* and states:

* Let x_i be the decision variable. The value of x_i is the destination state selected at stage i. For example, $x_2 = E$ means that at stage 2 the stand is thinned to reach state E at stage 3.

* Let $y_i(s, x_i)$ be the immediate yield of decision x_i at stage i and state x. For example, $y_2(B, E) = 145\ m^3/ha$.

* Let $Y_i(s, x_i)$ be the highest yield that can be obtained from stage i and all subsequent stages, given state s and decision x_i at stage i.

* Furthermore, let x_i^* be the decision that maximises $Y_i(s, x_i)$ and let $Y_i^*(s)$ be the corresponding maximum value of $Y_i(s, x_i)$.

Thus, $Y_i^*(s)$ is the highest yield from stage i and beyond, given state s at stage i. The object is to find $Y_1^*(A)$, the highest yield from stages 1, 2 and 3, given the initial stand state.

The dynamic programming , in this case, is based on *backward recurrence* (Buongiorno and Gilless, 1987). *Forward recurrence*, however, has been found to be more efficient and flexible than backward recurrence for thinning and rotation analyses (Brodie et al., 1978). With forward recurrence the optimal thinning regime and optimal rotation are simultaneously determined in a single run (Filius and Dul, 1992). Note that Nemhauser (1966) stated that, '... the direction (forward or backward) of multistage analysis may make a significant difference in the ease of solving the problem. The forward approach seems more natural, but the backward approach frequently leads more easily to the solution. Unfortunately, there appears to be a mental block against working backwards'.

Despite these different schools of thought, the basic difference between forward and backward recursion is that in the backward recursion the analysis

proceeds from stage N to stage one, and the optimal returns are found as functions of the stage outputs. In the forward recursion the analysis proceeds from stage one to stage N and the optimal returns are found as functions of the stage inputs.

Thus, the highest yield at stage 3 is determined first, for every possible state, and the corresponding best decision:

$$Y_3^*(s) \qquad \text{and} \qquad x_3^* \tag{IIA.1}$$

This is a trivial problem because there is only one possible decision for each state at stage 3. From this, the highest yield from stage 2 and stage 3, can be determined for each possible state at stage 2. That is, decisions x_2^* are found such that :

$$Y_2^*(A) = \max_{x_2} [y_2(A, x_2) + Y_3^*(x_2)] \tag{IIA.2}$$

From this, the highest yield from stages 1, 2 and 3 is determined, given the initial state at stage 1, that is, the decision x_1^* is found such that:

$$Y_1^*(A) = \max_{x_1} [y_1 A, x_1) + Y_2^*(x_1)] \tag{IIA.3}$$

$Y_1^*(A)$ is the highest total yield that can be obtained from the stand, and x_1^* is the best decision at stage 1. The decision x_1^* determines the best state at stage 2. For that state, the best decision x_2^* is known from the solution of (IIA.2). The decision x_2^* determines in turn the best state at stage 3. For that state it is found, from (IIA.1), that the best decision is x_3^*

APPENDIX III

DMISER3: Information for the user

DMISER3 is specifically written for discrete-time problems and was used successfully for generating thinning regimes for *P. patula* plantations for the very good sites in South Africa. DMISER3 is a useful tool for dealing with multistage optimisation problems and it employs Pontryagin's Maximum Principle for its solution technique. The optimisation problem in this document was a combined optimal parameter selection and optimal control problem. Thus the final crop number for a plantation was predicted in DMISER3 as a parameter. Further developments can be made by estimating the initial planting density as well.

It is always wise to use simple and differentiable state functions to ensure convergence to the true solutions. The program is well written in FORTRAN and it is very informative in its error reporting, which makes it easy to trace run time errors. Gradients have to be provided by the user, and the program always checks their validity by comparing them to their finite difference estimations. Any discrepancy is reported and the user has a choice of terminating a run and correcting the gradients, or continuing on. The manual (Jennings et al., 1990), however, is not all that helpful for a first timer. When DMISER3 has been successfully run, it is always wise to rigorously test the result by checking constraints satisfaction to the desired accuracy. This is important because there maybe hidden errors in the analytical formulation of the problem which may still result in seemingly successful execution of the program.

A typical problem that was faced in the thinning regime problem was the piecewise continuous function (4.5) for height estimation. If an optimal solution occurred at a point of discontinuity, then the program would have problems in deciding whether to take the upper or lower bound. DMISER3 is defaulted to make a choice and can even converge to a solution, but will report the partial derivatives of

that problem to be wrong from that point of discontinuity to the end of the control period. The problem was overcome by making the piecewise continuous height function continuous by applying smoothing techniques. At every point of discontinuity the following function was applied:

$$(b22 - b21)*h \tag{III.1}$$

where

b22 and *b21* represent the 2 functions at a point of discontinuity

$$h = \frac{1}{1 + exp(-10s_1)} \tag{III.2}$$

$$s_1 = \frac{x(1) - p}{p} \tag{III.3}$$

$x(1)$ = number of trees - stems/ha

p = point of discontinuity - stems/ha

This made the partial derivatives for the functions more complex, but not beyond the point of mathematical tractability.

A form of the discrete-time optimal control and optimal parameter selection problem is as follows:

$$\underset{(u,\, z)}{minimise} \quad \left\{ G_0(u, z) = f_0(x(M), z) + \sum_{t=0}^{M-1} g_0(t, x(t), u(t), z) \right\} \tag{III.4}$$

subject to the dynamics

$$x(t+1) = f(t, x(t), u(t), z), \quad t = 0, 1, ..., M-1 \tag{III.5}$$

with (possibly variable) initial conditions

$$x(0) = x^0(z) \tag{III.6}$$

Here M is a fixed positive integer, $x(t) = [x_1(t), ..., x_{n_s}(t)]^T \in R^{n_s}$, for $t \in I_M$, are the state variables, $u(t) = [u_1(t), ..., u_{n_c}(t)]^T \in R^{n_c}$, for $t \in I_{M-1}$, are the control variables and the

system parameters are $z = [z_1, ..., z_{n_z}]^T \in R^{n_z}$, which are independent of t. The control variables and system parameters are subject to simple bounds:

$$u_i^L \le u_i(t) \le u_i^U, \quad \forall t \in I_{M-1}, \quad i = 1, ..., n_c, \tag{III.7}$$

$$z_j^L \le z_j \le z_j^L, \quad j = 1, ..., n_z. \tag{III.8}$$

DMISER3 solves the above problem subject to a variety of additional constraints. Details on the types of constraints can be obtained from the user manual.

There are twelve subroutines of partial derivatives that have to be supplied by the user. Nine of these need to be evaluated at any time point in the interval, while the other three are only evaluated at the characteristic times of each constraint or the initial point. The following table indicates the functions and the gradients needed with the sizes of the Jacobian matrix indicated in the table entires.

Table III.1: The user supplied functions

	function	$\frac{\partial}{\partial x}$	$\frac{\partial}{\partial u}$	$\frac{\partial}{\partial z}$	evaluated at
g_0	1	$1 \times n_s$	$1 \times n_c$	$1 \times n_z$	any integer t
f	n_s	$n_s \times n_s$	$n_s \times n_c$	$n_s \times n_z$	any integer t
g	n_g	$n_g \times n_s$	$n_s \times n_c$	$n_g \times n_z$	any integer t
f	$n_g + 1$	$n_g + 1$ of $1 \times n_s$	O	$n_g + 1$ of $1 \times n_z$	$t=t_i$, $i = 0, ..., n_g$
x_0	n_s	O	O	$n_s \times n_z$	$t = 0$

The file *dframe.f* contains the underlying structure of the file of the user supplied functions. The user's task of constructing the file of user supplied functions is easily achieved by using a copy of *dframe.f* as the starting point. There are instructions embedded in *dframe.f* to make sure the order of elements of Jacobian matrices in the one dimensional arrays agree with what is expected by DMISER3. A copy of *dframe.f* for the thinning regime problem is shown below:

```
c File dframe.f
c_______1_______2_______3_______4_______5_______6__

      subroutine dusersetdat
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c
c If you want to include any user parameters which can then
c be varied at runtime then uncomment the following 3
c statements and include them in this and all the following
c routines.
c     integer nupar
c     doubleprecision upar(Pnupar)
c     common/ocupar/upar,nupar
c Note, however, that if you also want to include the file
c docsys.inc then leave the common block/ocupar/commented
c as it is included in the file docsys.inc.
c
c It is dangerous to uncomment the next include statement as
c the user must not use any of the variables used in this
c file. They are the global variables of the package.
c     include 'docsys.inc'
c
c The next include statement is the suggested method for the
c user to make any fixed parameters global to the user
c routines.
```

```
c     include 'docuser.inc'
c
c This routine is called once after docsetup and allows the
c user to set data into common blocks to create variables
c global to the user routines.
c The user should put user common blocks in the file
c 'docuser.inc' and then uncomment the include statement as
c appropriate.
      return
      end
c
c all the subroutines for the functions and derivatives
c follow.
c_______1_______2_______3_______4_______5_______6__

      subroutine docg0 (t,x,u,z,g0)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),g0
c Return the value of the objective function summand.
c
      g0=0.d0
      return
```

```
      end
c_____1_____2_____3_____4_____5_____6__

      subroutine docf (t,x,u,z,f)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),f(Pns),a1,b1,a2,b2,q,
     *a21,a22,a23,b21,b22,b23,ah1,ah2,s1,s2
c Return the value of the right hand side of the state
c equations.
c
      a1=0.93d0+1.d-5*x(1)-0.047d-6*x(1)**2+1.d-11*x(1)**3
      b1=2.32d0+4.24d-3*x(1)-0.354d-6*x(1)**2
      cc=1/sqrt((3.14159265358979/4.d0)/10000.d0)
      f(2)=a1*x(2)+b1
      q=sqrt(f(2)/x(1))
      a23=0.782d0
      b23=0.19d0+3.d-5*x(1)
      a22=0.85d0
      b22=0.095d0+4.d-5*x(1)
      a21=0.913d0
      b21=0.035d0+1.d-4*x(1)
      s1=(x(1)-400.d0)/400.d0
```

```
      s2=(x(1)-1000.d0)/1000.d0
      ah1=1.d0/(1.d0+exp(-10.d0*s1))
      ah2=1.d0/(1.d0+exp(-10.d0*s2))
      a2=a21+(a22-a21)*ah1+(a23-a22)*ah2
      b2=b21+(b22-b21)*ah1+(b23-b22)*ah2
      f(1)=x(1)-u(1)
      f(3)=a2*x(3)+b2*cc*q
      return
      end
c_____1_____2_____3_____4_____5_____6__

      subroutine docg (t,x,u,z,g)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),g(Pnga)
c Return either the value of the summands of the canonical
c constraints,
c or for continuous state constraints using eps-tau just
c supply the function
c values of h(t,x,u,z) describing the constraint, h(t,x,u,z) c >=0
c The computations involving eps and tau are done by DMISER3,
c not the user.
c
```

```
      g(1)=0.d0
      g(2)=x(1)-u(1)
      return
      end
c_____1_____2_____3_____4_____5_____6__

      subroutine docphi (ig,taut,x,z,phi)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer taut,ig
      real*8 x(Pns),z(Pnza),phi
c     Return the value of the ig-th phi function, i.e. one
c     scalar value.
c
      if(ig.eq.0)phi=-0.4*x(2)*x(3)
      if(ig.eq.1)phi=x(1)-z(1)
      if(ig.eq.2)phi=0.d0
      return
      end
c_____1_____2_____3_____4_____5_____6__
subroutine docxzero (z,x0)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
```

```
c     include 'docuser.inc'
      real*8 z(Pnza),x0(Pns)
c Return the value of the initial conditions of the state
c equations.
c
      x0(1)=z(2)
      x0(2)=0.d0
      x0(3)=0.d0
      return
      end
c Now follows the gradients wrt state variables.
c_____1_____2_____3_____4_____5_____6__

      subroutine docdg0dx (t,x,u,z,dg0dx)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),dg0dx(Pns)
c Return the gradient of objective summand wrt states.
c
      dg0dx(1)=0.d0
      dg0dx(2)=0.d0
      dg0dx(3)=0.d0
      return
```

```
      end
c_____1_____2_____3_____4_____5_____6__

      subroutine docdfdx (t,x,u,z,dfdx)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer  t
      real*8 x(Pns), u(Pnc), z(Pnza),dfdx(Pns*Pns),f(2), a1,
     *b1,a2,b2,da1,db1,q,db2,dq,a21,a22,a23,b21,b22,b23,ah1
     *,ah2,ds1,ds2,da2,db23,db22,db21,l1,l2,dah1,dah2
c Return the gradient wrt state of each rhs of the state
c equations as a row of dfdx. Note that dfdx is computed as a
c one dimensional array and that values are stored as if dfdx
c were dimensioned at dfdx(ns,ns). The order is df1/dx1,
c
c df2/dx1,...,dfns/dx1,df1/dx2,...,dfns/dx2,...,df1/dxns,...,
c dfns/dxns.
c
      a1=0.93d0+1.d-5*x(1)-0.047d-6*x(1)**2+1.d-11*x(1)**3
      b1=2.32d0+4.24d-3*x(1)-0.354d-6*x(1)**2
      da1=1.d-5-2.d0*0.047d-6*x(1)+3.d-11*x(1)**2
      db1=4.24d-3-0.354d-6*2.d0*x(1)
      f(2)=a1*x(2)+b1
      cc=1/sqrt((3.14159265358979/4.d0)/10000.d0)
```

```
q=sqrt(f(2)/x(1))
 a23=0.782d0
 b23=0.19d0+3.0d-5*x(1)
 db23=0.03d-3
 a22=0.85d0
 b22=0.095d0+4.d-5*x(1)
 db22=0.05d-3
 a21=0.913d0
 b21=0.035d0+1.d-4*x(1)
 db21=0.10d-3
 s1=(x(1)-400.d0)/400.d0
 s2=(x(1)-1000.d0)/1000.d0
 ds1=1.d0/400.d0
 ds2=1.d0/1000.d0
 ah1=1.d0/(1.d0+exp(-10.d0*s1))
 ah2=1.d0/(1.d0+exp(-10.d0*s2))
 a2=a21+(a22-a21)*ah1+(a23-a22)*ah2
 b2=b21+(b22-b21)*ah1+(b23-b22)*ah2
 dah1=ds1*10.d0*exp(-10.d0*s1)/(1.d0+exp(-10.d0*s1))**2
 dah2=ds2*10.d0*exp(-10.d0*s2)/(1.d0+exp(-10.d0*s2))**2
 da2=(a22-a21)*dah1+(a23-a22)*dah2
 l1=(b22-b21)*dah1+(db22-db21)*ah1
 l2=(b23-b22)*dah2+(db23-db22)*ah2
 db2=db21+l1+l2
dfdx(1)=1.d0
dfdx(2)=da1*x(2)+db1
```

```
      dq=0.5d0*((dfdx(2)/q)-q)/x(1)
      dfdx(3)=da2*x(3)+db2*cc*q+b2*cc*dq
      dfdx(4)=0.d0
      dfdx(5)=a1
      dfdx(6)=(a1*b2*cc)/(2.d0*sqrt(f(2)*x(1)))
      dfdx(7)=0.d0
      dfdx(8)=0.d0
      dfdx(9)=a2
      return
      end
c_____1_____2_____3_____4_____5_____6__

      subroutine docdgdx (t,x,u,z,dgdx)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),dgdx(Pnga*Pns)
c Return the gradient wrt state of each function in docg as a
c row of dgdx. Note that dgdx is computed as a one dimensional
c array and that values are stored as if dgdx were dimensioned
c at dgdx(ng,ns). The order is dg1/dx1,
c dg2/dx1,...,dgng/dx1,dg1/dx2,...,dgng/dx2,...,dg1/dxns,...,
c dgng/dxns.
c
```

```
      dgdx(1)=0.d0
      dgdx(2)=1.d0
      dgdx(3)=0.d0
      dgdx(4)=0.d0
      dgdx(5)=0.d0
      dgdx(6)=0.d0
      return
      end
c_____1_____2_____3_____4_____5_____6__
      subroutine docdpdx (ig,taut,x,z,dpdx)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
      integer taut,ig
      real*8 x(Pns),z(Pnza),dpdx(Pns)
      dpdx(1)=0.d0
      dpdx(2)=0.d0
      dpdx(3)=0.d0
      if(ig.eq.0)then
       dpdx(2)=-0.4*x(3)
       dpdx(3)=-0.4*x(2)
      endif

      if(ig.eq.1)dpdx(1)=1.d0
      return
      end
c_____1_____2_____3_____4_____5_____6__
```

```
      subroutine docdg0du (t,x,u,z,dg0du)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),dg0du(Pnc)
c Return the gradient wrt control of the objective summand.
c
      dg0du(1)=0.d0
      return
      end
c_____1_____2_____3_____4_____5_____6_

      subroutine docdfdu (t,x,u,z,dfdu)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),dfdu(Pns*Pnc)
c Return the gradient wrt control of each rhs of the state
c equations as a row of dfdu. Note that dfdu is computed as a
c one dimensional array and that values are stored as if dfdu
c were dimensioned at dfdu(ns,nc). The order is
c df1/du1,df2/du1,...,dfns/du1,df1/du2,...,dfns/du2,...,
```

```
c df1/dunc,...,dfns/dunc.
c
      dfdu(1)=-1.d0
      dfdu(2)=0.d0
      dfdu(3)=0.d0
      return
      end
c_____1_____2_____3_____4_____5_____6__

      subroutine docdgdu (t,x,u,z,dgdu)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),dgdu(Pnga*Pnc)
c
c
c Return the gradient wrt to control of each function in docg
c as a row of dgdu. Note that dgdu is computed as a one
c dimensional array and the values are stored as if dgdu were
c dimensioned at dgdu(ng,nc). The order is dg1/du1,
c dg2/du1,...,dgng/du1,dg1/du2,...,dgng/du2,...,dg1/dunc,...,
c dgng/dunc.
c
      dgdu(1)=0.d0
```

```
      dgdu(2)=-1.d0
      return
      end
c_____1_____2_____3_____4_____5_____6__
      subroutine docdg0dz (t,x,u,z,dg0dz)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),dg0dz(Pnza)
c Return the gradient wrt system parameters of the summand of
c the objective.
c
      dg0dz(1)=0.d0
      dg0dz(2)=0.d0
      return
      end

c_____1_____2_____3_____4_____5_____6__
      subroutine docdfdz (t,x,u,z,dfdz)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
```

```
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),dfdz(Pns*Pnza)
c Return the gradient wrt system parameters of each rhs of the
c state equations as a row of dfdz. Note that dfdz is computed
c as a one dimensional array and that values are stored as if
c dfdz were dimensioned at dfdz(ns,nz). The order is
c df1/dz1,df2/dz1,...,dfns/dz1,df1/dz2,...,dfns/dz2,...,
c df1/dznz,...,dfns/dznz.
c
      dfdz(1)=0.d0
      dfdz(2)=0.d0
      dfdz(3)=0.d0
      dfdz(4)=0.d0
      dfdz(5)=0.d0
      dfdz(6)=0.d0
      return
      end
c_______1_______2_______3_______4_______5_______6__

      subroutine docdgdz (t,x,u,z,dgdz)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer t
      real*8 x(Pns),u(Pnc),z(Pnza),dgdz(Pnga*Pnza)
```

```
c Return the gradient wrt system parameters of each function
c in docg as a row of dgdz. Note that dgdz is computed as a
c one dimensional array and that values are stored as if dgdz
c were dimensioned atdgdz(ng,nz). The order is
c dg1/dz1,dg2/dz1,...,dgng/dz1,dg1/dz2,...,dgng/dz2,...,dg1/
c dznz,...,dgng/dznz.
c
      dgdz(1)=0.d0
      dgdz(2)=0.d0
      dgdz(3)=0.d0
      dgdz(4)=0.d0
      return
      end
c_______1_______2_______3_______4_______5_______6__

subroutine docdpdz (ig,taut,x,z,dpdz)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      integer taut,ig
      real*8 x(Pns),z(Pnza),dpdz(Pnza)
c Return the gradient wrt system parameters of the ig-th phi
c function in dpdz.
c
      dpdz(1)=0.d0
```

```
      if(ig.eq.1)dpdz(1)=-1.d0
      return
      end
c_____1_____2_____3_____4_____5_____6__

      subroutine docdx0dz (z,dx0dz)
      implicit real*8 (a-h,o-z)
      include 'docpar.inc'
c     include 'docsys.inc'
c     include 'docuser.inc'
      real*8 z(Pnza),dx0dz(Pns*Pnza)
c Return the gradient wrt system parameters of the initial
c conditions of the state equations. Note that dx0dz is
c computed as a one dimensional array and that values are
c stored as if dx0dz were dimensioned at dx0dz(ns,nz). The
c order is dx1/dz1,dx2/dz1,...,dxns/dz1,dx1/dz2,...,
c dxns/dz2,...,dx1/dznz,...,dxns/dznz.
c
      dx0dz(1)=0.d0
      dx0dz(2)=0.d0
      dx0dz(3)=0.d0
      dx0dz(4)=1.d0
      dx0dz(5)=0.d0
      dx0dz(6)=0.d0
      return
      end
```

```
c Any other user defined code.

c_____1_____2_____3_____4_____5_____6__
```

DMISER3 input data file

The program DMKDATA generates a data input file for DMISER3. In the input file the following information is provided: file title, the size of the variables of the problem, number of states, control functions, system parameters, integration range of the objective function, stages (switching points) called knots in DMISER3, control bounds, bounds on the system parameters, the number of constraints and their type, accuracy and tolerance levels, optimisation tolerances for convergence etc. A copy of a typical input file is as follows:

```
%%%%%%%%%%%%%%  File Information  %%%%%%%%%%%%%%%%%%
File Title: volume production
Last Change: Mon Sep 19 13:18:42 1994
File Type: INPUT
DMISER3 Version:  1.0
Restart Info: EDIT  65, RESTART  0
%%%%%%%%%%%%%%  System Information  %%%%%%%%%%%%%%%%%%
ns =      3      (number of states)
nc =      1      (number of controls)
nz =      1      (number of system parameters)
tstart    5      (initial time)
tfinal    25     (final time)
lab = T      (objective or constraints use abs smoothing)
%%%%%%%%%%%%%%  Knot Sets Definition  %%%%%%%%%%%%%%%%%%
nksets =  1      (number of knot sets)
```

knot set # knot type number of knots

1 0 21

%%%%%%%%%%%% Knots for Type 1 Knot Sets %%%%%%%%%%%%%%%%

%%%%%%%%%%%% Control Definition %%%%%%%%%%%%%%%%%%%%

control # knot set bounded variation penalty

1 1 0.00000000D+00

%%%%%%%%%% Control lower initial upper %%%%%%%%%%%%%%%%%%

control # 1

1 0.00000000D+00 0.00000000D+00 7.00000000D+02

2 0.00000000D+00 0.00000000D+00 0.00000000D+00

3 0.00000000D+00 0.00000000D+00 0.00000000D+00

4 0.00000000D+00 0.00000000D+00 0.00000000D+00

5 0.00000000D+00 0.00000000D+00 0.00000000D+00

6 0.00000000D+00 0.00000000D+00 0.00000000D+00

7 0.00000000D+00 0.00000000D+00 0.00000000D+00

8 0.00000000D+00 0.00000000D+00 7.00000000D+02

9 0.00000000D+00 0.00000000D+00 0.00000000D+00

10 0.00000000D+00 0.00000000D+00 0.00000000D+00

11 0.00000000D+00 0.00000000D+00 0.00000000D+00

12 0.00000000D+00 0.00000000D+00 0.00000000D+00

13 0.00000000D+00 0.00000000D+00 0.00000000D+00

14 0.00000000D+00 0.00000000D+00 0.00000000D+00

15 0.00000000D+00 0.00000000D+00 7.00000000D+02

16 0.00000000D+00 0.00000000D+00 0.00000000D+00

17 0.00000000D+00 0.00000000D+00 0.00000000D+00

```
 18     0.00000000D+00  0.00000000D+00  0.00000000D+00

 19     0.00000000D+00  0.00000000D+00  0.00000000D+00

 20     0.00000000D+00  0.00000000D+00  0.00000000D+00

%%%%%%%%%%%%    System Parameters     %%%%%%%%%%%%%%%%%%

 No.      lower bound     initial value   upper bound

  1     1.50000000D+02  2.00000000D+02  8.00000000D+02

%%%%%%%%%%%%   Constraint Information   %%%%%%%%%%%%%%%%%%

ng =      2       (number of constraints)

ngeq =     1        (number of equality constraints)

ngineq =    1        (number of inequality constraints)

constraint # characteristic-time   smooth eps-tau   smooth absolute value

 1 equal          25              no              yes

 2 inequ          25              yes             yes

%%%%%%%%%%%%   Accuracy and Tolerance   %%%%%%%%%%%%%%%%%%

epsm =   2.22044605D-16 (machine accuracy)

reltest  1.00000000D+00 (relative error test)

epsjts   1.00000000D-02 (eps of eps-tau algorithm)

taujts   7.75000000D-02 (tau of eps-tau algorithm)

rhoabs   1.00000000D-02 (rho of abs smoothing)

%%%%%%%%%%%%   Optimisation Selection   %%%%%%%%%%%%%%%%%%

maxite=    60      (Maximum Number of Iterations)

maxfun=10(Maximum Number of Function Calls per Line Search)

 optprt=     2         (Optimisation reporting level)

 epsopt=  1.00000000D-05 (NLPQL accuracy for gradient, K-T)

 epscon=  1.00000000D-07 (NLPQL constraint accuracy)
```

%%%%%%%%%%% Input, Output and Error %%%%%%%%%%%%%%%%

error messages:

nout1 = 21 kout1= 1 fnout1= trial2.err

restart file:

nout2 = 22 kout2= 1 fnout2= trial2.res

solution file:

nout3 = 23 kout3= 0 fnout3= trial2.sol

tty out file:

nout4 = 6 kout4= 0 fnout4= stdout

save file:

nout6 = 26 kout6= 1 fnout6= trial2.sav

%%%%%%%%%%% Miscellaneous %%%%%%%%%%%%%%%%

ncheck = 999 (Frequency of user derivative check)

nsave = 5 (How often to save)

kabs = 1 (Which absolute value smoothing)

nupar = 0 (How many user parameters)

%%%%%%%%%%% End of Data File %%%%%%%%%%%%%%%%

A MatLab constrained optimisation program for South African P. patula

This same problem that was solved in DMISER3 was written as dynamic programming formulation in MatLab language and solved as a constrained optimisation problem. The only problems with the formulation was lack of flexibility for specifying constraints and other input variables, and comparatively longer execution times (i.e. on a 50 Mhz Macintosh IIfx with 20Mb RAM, it took 10-15 mins to get a solution as opposed to DMISER3 on the mainframe that took a few seconds for a similar problem). The MatLab function is as follows:

```
%-------------------------------------------------------------------------
%          CONSTRAINED OPTIMISATION FOR THINNING STRATEGIES
%-------------------------------------------------------------------------
%                INPUT  DATASCRIPT
for j=1:25              %Definition of rotation length
  u=0*ones(1,25);
  for k=1:10
    u(j)=k*100;
    g(j,k)=volume(u);  %An Optimisation subroutine call
  end
end
%-------------------------------------------------------------------------
%                OPTIMISATION SUBROUTINE
function [f,g]=volume(x)
g=-x;
%model for control purposes
%x is the control, amount of trees thinned, x>=0
```

```
%n is the number of trees in the plantation, density per ha

%y is the basal area, in m^2/ha

%h is the height in m

%v is the total volume in m^3/ha

%

%

%starting values here:

n(1)=1500;

y(1)=10;

h(1)=3;

%

%set horizon for simulation N

N=25;

%

%

%

%simulate over the horizon

for t=1:N

%

%number of trees at next year

n(t+1)=n(t)-x(t);

%

%basal area next year

%the functions a and b are separately defined

xx=n(t);

a1=0.93+0.01*xx/1000-0.047*(xx/1000)^2+0.01*(xx/1000)^3;
```

```
b1=2.32+4.24*xx/1000-0.354*(xx/1000)^2;

c=113;

if xx>1000, a2=0.782; b2=0.19+0.03*xx/1000; end

if (xx<=1000)&(xx>400), a2=0.85; b2=0.095+0.05*xx/1000; end

if xx<400, a2=0.913; b2=0.035+0.1*xx/1000; end

%

%

y(t+1)=a1*y(t)+b1;

%

% height next year

h(t+1)=a2*h(t)+c*b2*sqrt(y(t+1)/n(t));

%

end

f=-h(N+1)*y(N+1)*0.4;

%-----------------------------------------------------------------------------

%                    SIMULATOR

%This script will show the volume trends for what has been

%optimised in the above program

%model for control purposes

%u is the control, amount of trees thinned, u>=0

%x is the number of trees in the plantation, density per ha

%y is the basal area, in m^2/ha

%h is the height in m

%v is the stand volume in m^3/ha

%
```

```
%

clear

%starting values here:

x(1)=1500;

y(1)=10;

h(1)=3;

%

%set horizon for simulation N

N=25;

%

%

%for j=1:10

%for l=1:12-j

%for k=1:20

%for m=k+1:20

u= 0*ones(25);

u(15)=200;

u(16)=900;

%simulate over the horizon

for t=1:N

%

%number of trees at next year

x(t+1)=x(t)-u(t);

%

%basal area next year

%the functions a and b are separately defined
```

```
xx=x(t);

a1=0.93+0.01*xx/1000-0.047*(xx/1000)^2+0.01*(xx/1000)^3;

b1=2.32+4.24*xx/1000-0.354*(xx/1000)^2;

c=113;

if xx>=1000, a2=0.782; b2=0.19+0.03*xx/1000; end

if (xx<1000)&(xx>=400), a2=0.85; b2=0.095+0.05*xx/1000; end

if xx<400, a2=0.913; b2=0.035+0.1*xx/1000; end

%

%

y(t+1)=a1*y(t)+b1;

%

% height next year

h(t+1)=a2*h(t)+c*b2*sqrt(y(t+1)/x(t));

%

end


g((j-1)*10+1,(k-1)*20+m)=h(26)*y(26)*0.4;

hold off

clg

%plot(0.4*h.*y/10,'r')

end

end

end

end

%----------------------------------------------------------------------------
```

Index for keywords

—E—

—G—

—I—

—L—

—M—

—N—

—O—

—P—

—Q—

—R—

—S—

—T—

—U—

—W—

—Z—